AMIGA

11 Animation Products Reviewed

Amazing COMPUTING™

Your Original AMIGA™ Monthly Resource

Vol. 4 No. 1 Jan. 1989
US $3.95 Canada $4.95

Move over Mickey! Animation & Amiga

**Alive in 3D
A review of Caligari**

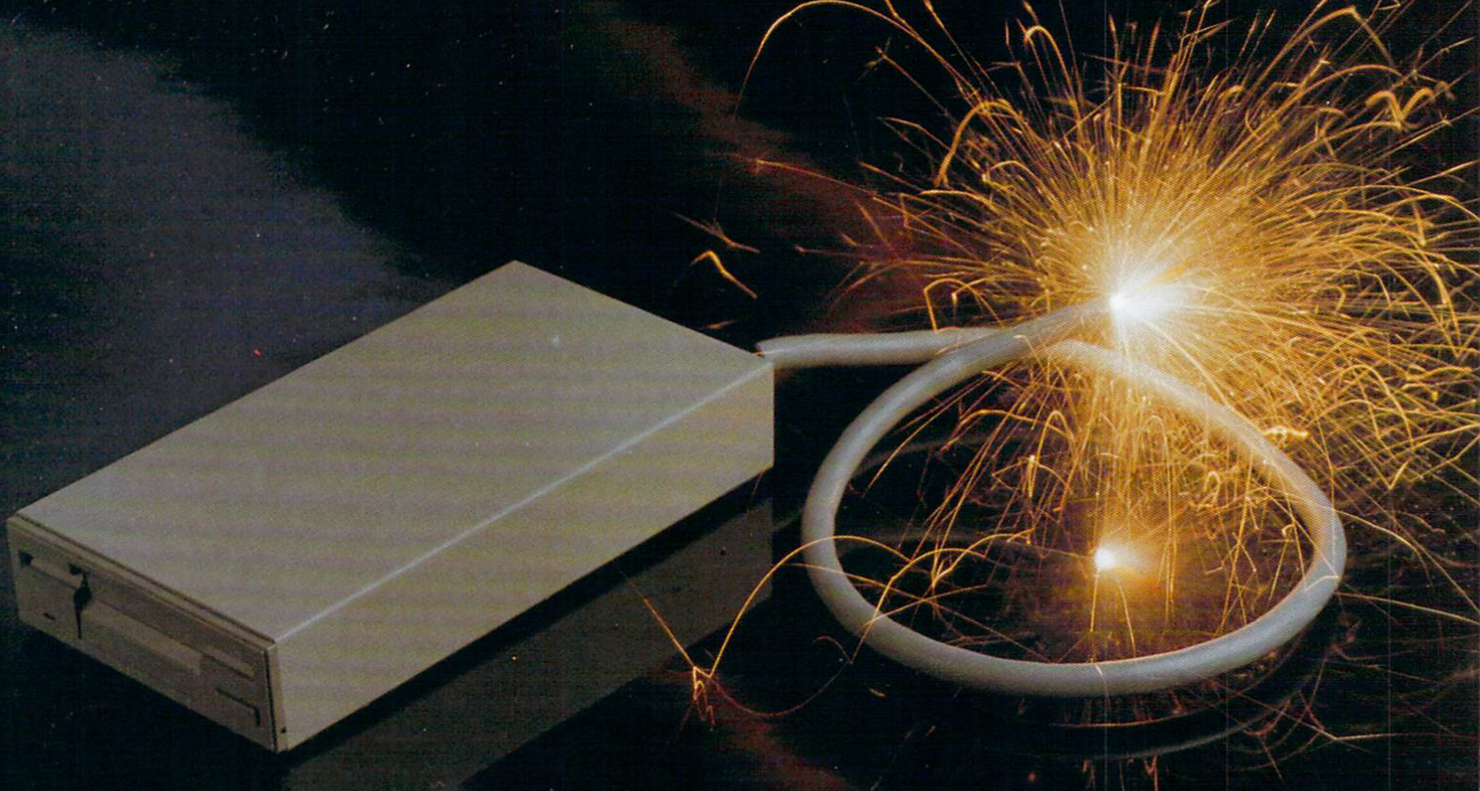
**AmigaBASIC
Industrial strength menus**

**In "C": Introduction to Pointers
: Scrolling super Bitmaps**

Plus much more

Second generation 2D animation software

Our DRIVE is Dynamite!



The AIR DRIVE

Dual-Sided, 880K
Color-coordinated chassis and faceplate
Smaller and lighter than the Amiga drive
External drive pass-through
Extra long 30" interface cable
Spring-loaded dust door
Compatible with all Amiga systems
One year warranty
Internal drive also available

CALL FOR DEALER NEAREST YOU.

MEGATRONICS

To Order: 800-232-6342

Ask about our
LOWEST PRICE GUARANTEE

INSIDE UTAH: (801) 752-2642
FAX: (801) 752-8752
CALL FOR A FREE CATALOG.
CREDIT CARDS VERIFIED FOR YOUR PROTECTION.

MEGATRONICS, INC. BOX 3660, LOGAN, UT 84321

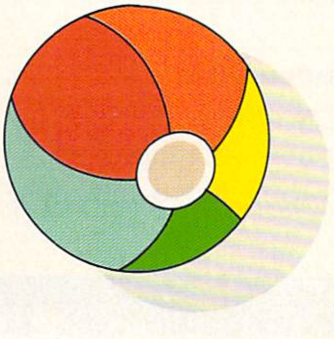
$79 **Small and powerful. The Avatex 1200E.**

Avatex MODEMS

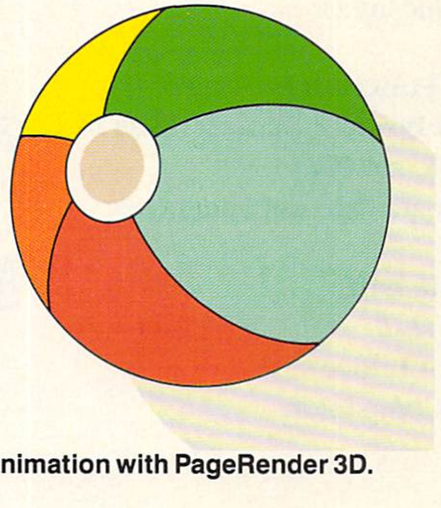
$105	1200P
$99	1200HC
$169	2400
$69	1200i
$159	2400i

FREE WITH EACH MODEM

Amiga communication software
& CompuServe access time.



- smooth, fast animation in any resolution
- powerful compression
- easy to add/edit static or dynamic backgrounds and foregrounds
- library of preset effects (transitions)
- create your own transitions easily
- interactive context-sensitive editor makes changes simple
- compact player program allows you to distribute your animations



Animates ALL IFF Images. Particularly Productive 3D Animation with PageRender 3D.

WHAT THEY SAY

Media: "This definitely looks like a very hot new animation package..."

Users: "PFX is wonderful!!!"

"This is an excellent program. The user interface is transparent, as it should be".

"The manual is well written... this is a truly great program, easy to use, friendly, and sophisticated perfect for our production needs".

"PFX is a great program".

"This is the first program I have found that will allow true cel animation".

Mindware International 110 Dunlop Street West, Box 22158, Barrie, Ontario, Canada L4M 5R3
For more information on the dealer nearest you! 1-705-737-5998 Order Line: 1-800-461-5441

Amiga is a trademark of Commodore Business Machines, Inc. Page Flipper Plus F / X is a registered trademark of Mindware International.

Amazing COMPUTING™

AMAZING PROGRAMMING

Death of a Process 71
by Mark Cashman
The polite multitasking guide to error handling in Modula-2

Industrial Strength Menus 77
by Robert D'Asto
Some snazzy submenus for your AmigaBASIC cuisine

Scrolling Through SuperBitMap Windows 101
by Read Predmore
How to implement SuperBitMaps for viewing or drawing into large graphic areas.

Pointers, Function Pointers, and Pointer Declarations in C 109
by Forest W. Arnold
Reducing data type dependencies

What's The Diff? 89
by Gerald Hull
A review of Lattice's Compiler Companion

Sync Tips 97
by Oran J. Sands III
Dot crawl, the Amiga and composite video devices.

C Notes From the C Group 98
by Stephen Kemp
Structures - A powerful feature of C

On the Crafting of Programs 94
by David J. Hankins
What Format is right for you

The Command Line 67
by Rich Falconburg
A look at new and improved Assembly Language commands

AMAZING DEPARTMENTS

Amazing Mail 6

Index of Advertisers/ Reader Service Card 112

Public Domain Software Catalog 121

Your Original AMIGA Monthly Resource

•TABLE OF CONTENTS•

Volume 4, Number 1
January 1989

AMAZING ANIMATION

How May I Animate Thee? Let Me Count The Ways 10
by Shamms Mortier
An overview of animation techniques.

Alive in 3D 15
by Shamms Mortier
A review of Caligari, a high-end 3D sculpting and animation package.

Stop-Motion Animation On The Amiga 23
by Brian Zupke
A hands-on approach to animation on the Amiga.

The Wonderful World of Hashnique 25
by Shamms Mortier
A review of the Amiga software products of Hash Enterprises

Second Generation 2D Animation Software 32
by Geoffrey Williams
Cel Animators and Key Frame Animators, how they differ and a look at their use.

AMAZING REVIEWS & REPORTS

STELLARYX Review 42
by Stephen Kemp
Exciting & challenging! Terrific stereo and sound effects

Arkanoid Imposters: Unmasking the impostors 39
by Jeffery Scott Hall
A look at Arkanoid look-alikes

Questron II 44
by Jeffery Scott Hall
Questron II - A journey back in time

Las Vegas Comdex Report 47
by Louise Brinkmann
Commodore's new 2500, 2500 UX, and more!

Philadelphia World of Commodore 50
by Chris Darsch & Rick Rae
Highlights of Philadelphia's Commodore Show

Roomers 58
by The Bandito
Commodore's deal, RAM chip crisis, and more!

In this competitive world there is only one winner.

The results are in and Hurricane has won. According to the August AmigaWorld: "Currently, the Hurricane board is the best way to increase the performance of your Amiga."

Now Ronin has designed a whole family of Hurricane products that will make your Amiga 1000 or 2000 run up to 20 times faster. You'll save hours on ray tracing, desktop video, animation, and other math intensive applications.

The Hurricane family is fully modular, so you can start for as little as $600 and upgrade in stages. Start racing with the 68020. You can then add a 68881. Later move up to our 68030 and maybe the 68882. Add up to 4 megabytes of 32-bit memory and your Amiga will sprint faster than a VAX minicomputer.

Why waste time and money on less performance. Run on in to your favorite dealer and get a Hurricane, or call us for more information. But whatever you do, don't be left at the starting line while everyone else passes you by.

HURRICANE™ *Dealer inquiries invited*

Ronin

P.O. Box 1093, Alameda, CA 94501 U.S.A
Telephone (415) 769-9325
Fax (415) 865-9852 Twx 9102502720 Ronin UD

Amiga is a trademark of Commodore-Amiga, Inc. 68020, 68030, 68881, 68882 are trademarks of Motorola. Hurricane is a trademark of Ronin Research & Development Inc. VAX is a trademark of Digital Equipment Inc.

Amazing COMPUTING®

ADMINISTRATION

Publisher: Joyce Hicks
Assistant Publisher: Robert J. Hicks
Circulation Manager: Doris Gamble
Asst. Circulation: Traci Desmarais
Asst. Circulation: Donna Viveiros
Corporate Trainer: Virginia Terry Hicks
Traffic Manager: Robert Gamble
International Coordinator: Marie A. Raymond
Marketing Manager: Ernest P. Viveiros Sr.

EDITORIAL

Managing Editor: Don Hicks
Editor: Ernest P.Viveiros Jr.
Hardware Editor: Ernest P. Viveiros Sr.
Music & Sound Editor: Richard Rae
Copy Editor: Michael Creeden
Copy Editor: Amy Duarte
Art Director: William Fries
Photographer: Paul Michael
Illustrator: Brian Fox
Production Manager: Donna M. Garant

ADVERTISING SALES

Advertising Manager: Alicia Tondreau
Marketing Assistant: Melissa J. Bernier

1-508-678-4200
FAX 1-508-675-6002

SPECIAL THANKS TO:
Buddy Terrell & Byrd Press
Betsy Piper at Tech Plus
Bob at Riverside Art, Ltd.
Swansea One Hour Photo

Amazing Computing™ (**ISSN 0886-9480**) is published monthly by PiM Publications, Inc., Currant Road, P.O. Box 869, Fall River, MA 02722-0869.

Subscriptions in the U.S., 12 issues for **$24.00**; in Canada & Mexico surface, **$36.00**; foreign surface for **$44.00**.

Application to Mail at Second-Class Postage Rates pending at Fall River, MA and additional mailing offices.

POSTMASTER: Send address changes to PiM Publications Inc., P.O. Box 869, Fall River, MA 02722-0869. Printed in the U.S.A.

Copyright© Nov. 1988 by PiM Publications, Inc. All rights reserved.

First Class or Air Mail rates available upon request. PiM Publications, Inc. maintains the right to refuse any advertising.

Pim Publications Inc. is not obligated to return unsolicited materials. All requested returns must be received with a Self Addressed Stamped Mailer.

Send article submissions in both manuscript and disk format to "Submissions, C/O PiM Publications, Inc. Requests for Author's Guides should be directed to the address listed above.

Amazing Dealers

The following are Amazing Dealers, dedicated to supporting the Commodore-Amiga™. They carry Amazing Computing™, your resource for information on the Amiga™. If you are not an Amazing Dealer, but would like to become one, call PiM Publications, Inc.:

1-508-678-4200

Alaska	
That Computer Place	Anchorage
Juneau Electronics	Juneau
Alabama	
Abax Data Systems	Huntsville
Alabama Book Store	Tuscaloosa
Command Computers	Birmingham
Madison Book & Computer	Madison
Mel's Photo & Computer Shop	Montgomery
The Computer Nut	Birmingham
The Computer Image	Birmingham
Universal Computer System	Mobile
Arkansas	
Commodore Connection	Sherwood
Microtronix	Fort Smith
The Micro Shop	Little Rock
Arizona	
Books Etc.	Tempe
Books Brothers Ltd.	Tucson
Computer Libray, Inc.	Tempe
Computer Works	Glendale
Copperstate Business Systems	Phoenix
North American Digital	Tucson]
Try Tech Inc.	Pheonix
California	
3 E Software And Systems	Hayword
American Mags & Books Ltd.	Englewood
Augnet	Fremont
Balboa Office Supply	San Diego
Bits & Bytes	Fairfield
Bits n Pieces Comp Prod.	Anaheim
Bookland	Marysville
Brown Knows Computing	Redlands
CAL-Tech Bookstore	Pasadena
Candy Computer	Elk Grove
Centerfold News	Los Angeles
Central Park Bookstore	San Mateo
Century Computer Systems	La Habra
Circus of Books	Los Angeles
Circus of Books	Los Angeles
Commodore Land	Alhambra
Commotech	National City
Computer Attic	Palo Alto
Computer Literacy	Santa Clara
Computer Literacy Books	Sunnyvale
Computer Nook, Inc.	San Bernadino
Computer Outlet	San Diego
Computer Spectrum	Burlingame
Computer Time	Citrus Hts
Coronet Newsstand	Oceanside
Covina Bookstore	Covina
Dataphile	Watsonville
Delaurer's News Agency	Oakland
Discount Computer Station Inc.	Long Beach
Empire Computers	Santa Rosa
Encinitas News	Encinitas
ECX Computer Co.	Walnut Creek
Fisher Computers + Software	Rancho Cucamonga
Harding Way News	Stockton
Heinrichson, Schneider & Young, Inc.	Berkeley
Home Computing Center	Concord
Home Computing Center	Mill Valley
Home Computing Center	San Leandro
HT Electronics	Sunnyvale
International Telecine	Santa Ana
K J Computers	Granada Hills
Little Professor Books	Placentia
LTC Computers	Santa Ana
Lyon Enterprises	Berkeley
Micro Flight	San Jacinto
NewsBeat	Davis
Northridge Newsstand	Northridge
Op Amp Tech Books	Los Angeles
Paras Newsstand	San Diego
Put's Electronics	North Highlands
R&R Software	Chula Vista
Ridgecrest Computer Ctr. Inc.	Ridgecrest
Sawyer's News Inc.	Santa Rosa
Software First	Santa Rosa
Software Cellar	Anahiem
Software Central	Pasadena
SoftwareLand	San Diego
Software Plus	Citrus Hieghts
Software Solutions Plus	El Cerito
S.O.S. Computers	Los Angeles
Stacy's #5	Modesto
Technitronics Service	South Pasadena
The Booksmith	San Francisco
Third Avenue News	Chula Vista
Tower Books	Chico
Tower Books	Citrus Hts.
Tower Books	Concord
Tower Books	Mountain View
Tower Books	Sacremento
Tower Books	San Mateo
The Computer Room	Scotts Valley
Universal News	Hollywood
WCI Computer Central	Eureka
Warner Engineering	San Diego
Winner's Circle	Berkley
World Book & News	Hoolywood
Colorado	
Aurora Newsstand	Aurora
Citadel Computers	Colorado Springs
Computer Discount	Denver
Computer Reports	Boulder
Computer Room	Aurora
Micro Computer World	Fort Collins
Micro World	Lakewood
Sun Country Computers	Colorado Springs
Connecticut	
Aetna Life Club Store	Hartford
Amigo Business Computer	Norwalk
Computer Outlet	PlaVernon
Computer & Software Connections	Plantsville
Demic Electronics	Hamden
Friendly Computers	Plantsville
Mnemonics Inc.	Norwalk
One Byte	Quaker Hill
Personal Computer Center	Norwich
Software City	Orange
Software Kingdom	East Hartford
Tricom Computers	North Windham
District of Columbia	
Programa	Washington
Delaware	
Castle Video	Newark
Delaware Electronics	New Castle
Newark Newsstand	Newark
Florida	
A A Computers	Jacksonville
Chuck's Newsstand	Vera Beach
City Newsstand	Melbourne
Commodore Country Inc.	Pinellas Park
Computer Bar	Pennsacola
Computer Base	Brandon
Computer Image	Miami
Computer Terminal Inc.	Sebring
Computers + Video Creations	Palm Beach
Computers 4 Rent	W. Palm Beach
Computers Plus	S. Daytona
Computer Room	Ft. Lauderdale
Cotronics Inc.	Stewart
Creative Equipment	Miami
Education Computers Etc.	Talahasse
Family Computers	Sarasota
Florida Book Store	Gainesville
Gulf Coast Computer Exchange	PanamaCity
Heartland Computers	Daytona Beach
Joe's News, Inc.	MiamiMicro Computer
Center	Largo
Newton's Video	Boca Raton
PC Collection	Coral Gabels
Program Store	Clearwater
Prospect News & Books	Ft. Lauderdale
Random Access Computers Inc.	Ft. Walton Beach
RGB Computers and Video Creations	Palm Beach Gardens
Sunny's At Sunrise, Inc.	Sunrise
Computer Room	Ft. Lauderdale
The Open Door	Merritt Island
Georgia	
BFJ Enterprises	Roswell
Cardinal Computers	Dalton
Software South Inc.	Savannah
The 64 Store	Atlanta
Hawaii	
Astro Computers & Electronics	Aiela
Computronics	Honolulu
Data-One Computer Center	Honolulu
J & C Repair	Aiea
Idaho	
A.B.I. Computer & Video	Nampa
Boise Computer & Video Inc.	Boise
Illinois	
Audio Visual Req.	Chicago
Computer Software Connection	Moline
DigitalWorld	Addison
DOS Computer Center	Niles
I.C.V. Data Center Inc.	Mundelein
Keep In Pace Computers	Champaign
M2J2 Inc.	Champaign
Kroch's & Brentano's	Chicago
Kroch's & Brentano's	Schaumburg
Kroch's & Brentano's	Orland Park
Kroch's & Brentano's	Auroa
Micro Age	Oaklawn
Micro Ed Enterprises Inc.	Chicago
Micro Tek	Kankakee
Ring Software	Geneva
Software City	Bellville
Software Plus	Wheeling
Software Plus Chicago	Chicago
Software Plus West	Hanover Park
The Memory Expansion	Bloomington
Unique Computer	Sterling
Warren Radio/FRANCO	Peoria
Indiana	
Bookland	Indianoplis
Burkat Computer Ctr.	South Bend
Bytrex of Angola	Angola
Computer Corner	Marion
Computer People Inc.	Michigan City
Computer Products Unlimited	Ft. Wayne
CPU Inc.	Indianapolis
Indiana News	Indianapolis
Micro Computer Centers Inc.	Indianapolis
Reese Equipment Co.	Plymouth
The Book Corner	Bloomington
The Computer Room of Merriville	Merriville
Vons Computers	West Lafayette
Iowa	
Gronet Computers	Des Moine
Micro Computer Applications LTD	Marshaltown
Soft Search	Ames
Kansas	
Arc Electronic Distributors	Wichita
J & L Electronics	Liberal
MidKansas Computers	Newton
Midwest Computers	Manhattan
Software City	Overland Park
Kentucky	
Computer Dimensions Inc.	Lexington
Computer Mania Inc.	Florence
Computer Store of Kentucky	Louisville
Hawley Cooke	Louisville
Reality World Computers	Lexington
Louisiana	
City Newstand	Baton Rouge
Computer Time	Shreveport
Lakeside News	Metairie
Modem Business Machines	Metairie
Software Center International	Metairie
Software Source	Lafayette
The Book Shelf	Slidell
The Computer Clinic	Lafayette
Maine	
Computer Barn	Damariscotta
Hands On Computing	Westbrook
Radio Shack Associate Stores	Jay
Skylight Software	Belfast
Maryland	
Buried Treasure	Rockville
Cal Com Inc.	Silver Spring
Compuvision Vision Center	Columbia
Greetings and Readings	Towson
Knowledgeware Systems	Kensington
Micro Computer Center	Baltimore
Professional Micro Services	Baltimore
Software Advantage	Rockville
Software N Things	College Park
Waldorf Computer	Waldorf
Massachusetts	
Computer Zone	North Attleboro
HCS Computer Center	Marshfield
LCA Video & Computer Ctr.	Norwood
Memory Location	Wellsley
OmniTek Computers	Tewksbury
Pioneer Valley Data Equipment	Amherst
Soft Designs	Watertown
Softpro	Burlington
The Bit Bucket	West Newton
The Software Haus	Quincy
Tycom Inc.	Pittsfield
Michigan	
CompuSoft	Kalamazoo
Computer Emporium	Lansing
Direct Access	Lansing
Central Park Computers	Okemos
Global Computer Center	Dearborn
I/O Software	Flint
Michigan News Agency	Kalamazoo
Michigan Software	Novi
Micro World	Livonia
Micro World	Milford
Micro World	Roseville
Micro World	Southfield
Micro World	Southgate
Micro World	Sterling Heights
Microworld Computer Center	Dearborn
Midland Computer Shop	Midland
Norman Business Control Sys.	Wyoming
Readmore Newscenter	Lansing
Rochester Book Center	Rochester Hills
Roseville Computer Store	Warren
Slip Disk	Madison Heights
Software City	Grand Rapids
Specialists Computer	Livonia
State Street Computers	Ann Arbor
T.J. Enterprises	Clinton
The Edge Connector	Clio
United Computer	Canton
Ye Olde Computer Shoppe	Ypsilanti
Minnesota	
Computer Excercise World	Minneapolis
Computer Stuff	Minneapolis
Computers, etc.	Eagan
JMH Software	Maple Grove
Lookahead Software Compute	rNorthfield
Shinder's	St. Paul
Shinder's	Maplewood
Shinder's	Burnsville
Shinder's	Minnetonka
Shinder's	Edina
Shinder's	Crystal
Shinder's Read More	Minneapolis
Specialists IN	Hopkins
Valiant Inc.	Stillwater
Mississippi	
Computer Connection	South Haven
National Computer Center	Ocean Springs
Missouri	
Associated Computer Services	Springfield
Brands Mart Computers	Kansas City
Carthage Electronics	Carthadge
Data Byte Computer Co.	St. Louis
Software City	Independence
Systems Plus	Overland Park
Systems Plus of St. Louis	St. Louis
Montana	
Appelgren Computer Systems	Great Falls
Nebraska	
Double E Electronics	Omaha
My Com Inc.	Plattsmouth
Nevada	
Century 23 Inc.	Las Vegas
Computer World	Las Vegas
Tower Records	Las Vegas
New Hampshire	
Dartmouth Bookstore	Hanover
Diversified Computers	Keene
New Jersey	
Business System Professional	Scotch Plains
Family Computer Centres	South Orange
Morris County Stationers	Chester
Software Station	Rockaway
Take One More Byte	Hoboken
The Program Store	Eaton Town
The Program Store	Wayne
Unlimited Software Inc.	Rutherford
New Mexico	
New Horizon Computer System	Alamogordo
Page One Newstand	Albuquerque
Road Runner	Espanola
Technological Concepts	Las Cruces
New York	
Amuse	New York
Amicus- Hudson Valley	Wappingers Falls
Amigo Business Computer	East North Port
Byte Shop	Merrick
Compurite Center	White Plains
Computer Outlet	Jamestown
Computers Etc.	Syracuse
Computerware	East Meadow
Contemporary Computing	Latham
Creative Computers	Johnson City
Cromland Inc.	Brooklyn
Leeder's Video	Schenectady
Leigh's Computers	New York
MCP Associates	Astoria
Mnemonics	North Port
Readers Market	Tonawanda
Software & Such	Scotia
Software City	Forest Hills
Software Link	White Plains
Software Supermarket	Tonawanda
Star Tech Systems	Massena
Synfaire Trading	New York
Sysput Computers	Brooklyn
Tadiran	New York
Tedrow Business Products	Rochester
The Computer Cellar	Albany
Unicom Electronics	Johnson City
Video Computer Bin	Brooklyn
Video Computer Center	Rome
Village Computer Inc.	New York
World Computers Inc.	Hicksville
North Carolina	
Digitz	Raleigh
K&S Newsstand	Clemmons
K&S Newsstand	Kernersville
K&S Newsstand	Winston-Salem
News Center	Cary
Parker News Inc.	High Point
Software City	Raleigh
Software City	Charlotte
TDS Computers	Durham
Triad Computers	Winston-Salem
North Dakota	
Computer Associates	Fargo
G. F. Computers	Grand Forks
Software House	Fargo
The Computer Store	Minot
Ohio	
B5 Software	Columbus
Computer Network	Newark
Computer Site	Cleveland
Earthrise Micro Systems	Columbus
Eddy's Cigar Store	Elyra
Fairborn Home Computer	Dayton
Lakewood International News	Lakewood
Lakes Consumer Electronics	Akron
Leo's Book & Wine Shop	Toledo
Microtech Computer Center Inc.	Toledo
Microware Magic	Fairfield
News Shop	Kent
North Coast Programming	Willoughby
Paper & Ink	Tiffin
Softwaire Centre International	North Olmsted
Software and More	Cincinnati
Software Connection	Mayfield Hts.
Software Plus	Columbus
Somers Photo & Comp.	Cleveland Heights
Specialist Computer System	Toledo
Oklahoma	
Little Proffessor Books	Stillwater
Merit Computers	Oaklahoma City
Micro Link Computer Store	Tulsa
P C Tech	Tulsa
Second Hand Software	Oklahoma City
Video-Comp Inc.	Lawton
Oregon	
Andromeda	LaGrande
Clackamas Computers	Beaverton
Clackamas Computers	Clackamas
Comm-Shack	Salem
Computron Business Systems	Portland
PC Express	Eugene
Powell's Tech	Portland
Rich Cigar Store	Portland
Sixth & Washington News	Portland
Software Pipline	Beaverton
Tech Books	Portland
Pennsylvania	
Alpha Omega EBE	Warren
ATD Software	Sayre
Basic Computer Systems	Erie
Basic Computer Systems	Hermitage
Circle Computer	Ephrata
Computer Solutions	Newtown
Computer Wurx	Stroudsburg
Computers and Games	Reading
Downington Computer Center	Downington
East Coast Software	Hershey
Floppy Disk	Mechanicsburg
Floppy House Software	Palmyra
Gene's Books	King of Prussia
Micro Master Inc.	Monaca
North Central Digital Systems	Danville
Pittsburgh Computer Store	Pittsburgh
Software Hut Inc.	Philadelphia
Tasco Computer Center	Lancaster
Triangle Computers	Indiana
Walden Hill Images	Berlin
Rhode Island	
Micro Limits	Smithfield
Micro Limits	Greenville
Software Connection	Warwick
South Carolina	
Avcom International Inc.	West Columbia
Micro Computer Depot	Sumter
Watson's Jewlery	Chesterfield
Software Haus	Charleston Hts
South Dakota	
Computer Business Center	Rapid City
Tennessee	
ComputerLab of Memphis	Memphis
Computer Hardware Services	Nashville
Data International	Hickson
The 64 Store	Bartlett
Texas	
All Systems Go Software	San Antonio
Amazing Computer Systems	Carrollton
Bookstop	Houston
Bookstop	San Antonio
Byte Shop	Austin
C.E.M. Corporation	Brazoria
Colonel Video	Houston
Computer Age	Houston
Computer Age	Dallas
Computer Magic	Austin
Computer Skills	Eules
FNS Computer Services	Texarkana
IMaxwell Book Inc.	Desota
Metropolitan Computer	Richardson
Micro Search	Houston
Software Library	Witchita Falls
Software Terminal	Fort Worth
The Computer Experience	San Antonio
The Computer Stop!	Amarillo
The Federated Group	Mosquite
Utah	
Magazines and Gifts	Salt Lake City
People's Computers	Provo
Vermont	
Ray Supply Inc.	South Burlington
Virginia	
Family Computer Center	Fairfax
Family Video & Computer Center	Newport News
I/O Computer	Norfolk
Software Central	Norfolk
Software City	Richmond
Tract Software	Prince William
Virginia Micro Systems	Woodbridge
Washington	
Amazing Electronics	Spokane
A.P.P.L.E. CO-OP	Renton
Apple Valley Computers	E. Wenatchee
Bellingham Computer Center	Bellingham
Bits, Bytes and Nibbles	Spokane
Bull Dog News	Seattle
Byte World	Kent
Computer Exchange Services	Bellingham
Computer Quest	Edmonds
Computers + Inc.	Tacoma
Family Computers	Everett
Micro Age Computer Store	Bellevue
Nybbles & Bytes	Tacoma
Omni International	Seattle
Port Book and News	Port Angeles
Programs Plus	Seattle
Real Time Systems	Seattle
Take A Byte	Kennewick
Tower Books	Bellevue
Y Not Magazines and Boks	Federal Way
Wisconsin	
20th Century Books	Madison
Computer Software Center	Milwaukee
Fox Valley Personal Computer	Oshkosh
Madison Programmer's Guide	Madison
Mayfair Computers	Milwaukee
Mom's Computers	Stevens Point
River City Software	La Crosse
Software City	Racine
TMW Software Inc.	Wausau
Wyoming	
Byte Emporium	Laramie
Computer Works	Laramie
Crazy Mountain Electronics	Cooper
Canada	
Compute or Play	Alberta
Computerworks	Alberta
Computer Shop	Alberta
Data Pro	Alberta
Computer Shop of Calgary	Alberta
The Software House	Alberta
Hindsin Computing	Alberta
Red Deer Micro Systems	Alberta
Memory Land	Alberta
Compu Care	British Columbia
Computer King	British Columbia
Conti	British Columbia
Nutec	British Columbia
Pacific Business Technologies	British Columbia
Sprita	British Columbia
Strider	British Columbia
Super Software	British Columbia
Canadian Computer Era	Manitoba
Micromart	Manitoba
ComputerWorld	Manitoba
J.W. Allen	Manitoba
Kobetak	Manitoba
Minicomp	Manitoba
Arkon	Ontario
Chatam Computer	Ontario
Computerland	Ontario
Comspec	Ontario
Electronics 2001	Ontario
Hastings Data	Ontario
The Computer & You	Ontario
Thornhill	Ontario
Software Co.	Quebec
Deltronie Solare	Quebec
Mierodata	Quebec
Info Mereier	Quebec
Laval Micro	Quebec
Vision Tronique	Quebec
Distribution Visipro	Quebec
Distributed in Canada by	
Ingram Software Distributors	Concord, Ontario
Compulit Distributors	British Columbia
Phase 4 Dist. Inc.	Calgary, Alberta
Argentina	
Logic Computer	Buenos Aires
Australia	
Computermart	Leederville,WA
Diskworks	Greenwich, NSW
John Verhoeven	Cloverdale, WA
Maxwell Office Equipment	Abbotsford, Melburn
Run Soft	Melbourne
France	
A.M.I.E.	Paris
Saudia Arabia	
Abdulla Fouad & Sons	Daman
Scotland	
The Amiga Center of Scotland	Edinborough
Sweden	
Nicholas Peerson	Lund
W. Germany	
Greenville Trading & Import	Oberursel

Amazing Computing™ is also available in most B. Dalton Booksellers, B. Dalton Software Stores, Crown Books, Software Etc., selected WaldenBooks Stores, and Walden's Software Store locations.

AMAZING MAIL

Dear AC

You published my letter of reply to the Roomers column in this past November issue. Something got lost either in my printout or your editing. I didn't save a copy of the letter I sent. The last line reads— "two 880 floppies and two meg 3 1/2" miniscribe hard drives.

It should have read "and two 32 meg, 3 1/2" miniscribe hard drives."

Don Morgan
Parsippany, N.J.

Thank you for the correction and the quick response.

Dear AC

I would like to have our User Group published in your magazine. I have respect for Amazing, and trust that your readership will be happy to learn of our Chicago area group. Unfortunately, our schedule does not permit meeting dates such as "the fourth Wednesday", so I hope you can publish at least some of our meeting dates. Below is our announcement that I hope you can print as soon as possible.

Hello from the Chicago Northwest suburban Amiga User's Group, or CNSAUG (pronounced sin-sawg). We meet monthly, and provide a monthly Public Domain software disk. We currently list about 130 members, and can always use additional support from Amazing readers who want to participate! We hold Special interest Groups (SIGS), and will soon hold one for 1.3. We are also dues-less.

Officers:
  Tom Karlmann-Pres. (312) 934-3334
  Don Vanni----Librarian (312) 478-7362
Meeting Place:
  Arlington Heights Public Library
  500 North Dunton Ave.
  Arlington Heights, IL
Meetings 1989:
  (from 7:00 pm to about 10:00 pm)
  Jan 25, Feb 22, Mar 23, Apr 20, May 18, Jun 22, Jul 26, Aug 30, Sep 27, Oct 23, & Nov 29
Mailing Address:
  CNSAUG c/o Thomas Karlmann
  4437 Westbridge CT
  Hoffman Estates, IL 60195

I thank all the Amazing people for their Amazing magazine, I have an amazingly good time reading it, now join our amazing user's group and publish my amazing notice—I had to get that out of my amazing system!

Amazingly,
Thomas Karlmann

I think you will be amazed at how much fun you will have.

Dear AC,

All of the letters I've seen in Amazing Computing and other Amiga magazines that deal with modifying the startup-sequence to copy frequently used commands to RAM:c also require that C: be reassigned to RAM:c. This is fine until you need to execute a command that is not in RAM:c. In this case you must either assign C: back to SYS:c or type in a full path name.

A better way which avoids all this extra typing is to add RAM:c to the search path in addition to paths for RAM: and any other directories you want searched. Note that the RAM: path does not cause RAM:c to be searched.

Add the following to the list of paths in your startup-sequence:

```
Path RAM:c add
Delete any reassignment of C:
```

After you save your changes and reboot, the search path sequence will be:
  The current directory (first)
  The paths you added in the order you added them
  C: (last)

Now you can type commands without worrying about where they are.

Sincerely,
Spence Blakely
Portsmouth,RI.

Dear AC:

While working on an error trapping routine in AmigaBasic I attempted to trap error 74, which in the Amiga Basic manual is listed as "Unknown Volume". The trap failed and it took a few minutes to discover that the errors that I should have been trapping were in the manual!

Although error 74 is listed on page A-11 as the code for "Unknown Volume", when you run "error 74" in direct mode you get the message "Rename across disks", which is not listed in the manual. Since there are a possible 255 error messages, I wrote the following bit of code to find which error number corresponds to "Unknown Volume".

By "inputting" a volume which wasn't in a disk drive, I caused a requester to appear asking for the named volume. Cancelling the requester generated an "Unknown Volume" error which was then trapped and its error code (49) was printed on the screen. Running "error 49" in direct mode verified that it is the proper code number for "Unknown Volume".

Obviously, the third and fourth line of the program can be changed to simulate other errors to verify the manual's error codes.

```
ON ERROR GOTO 10
DIM X (255)
INPUT FILE$
OPEN FILE$ FOR INPUT AS #1
10: FOR X=0 TO 254
X=X+1
IF ERR=(X) THEN PRINT (X)
NEXT X
PRINT "DONE"
END
```

I hope this is helpful to anyone who might have a problem with AmigaBasic error codes.

Yours Truly,
John M. Salimbene
Columbus, OH

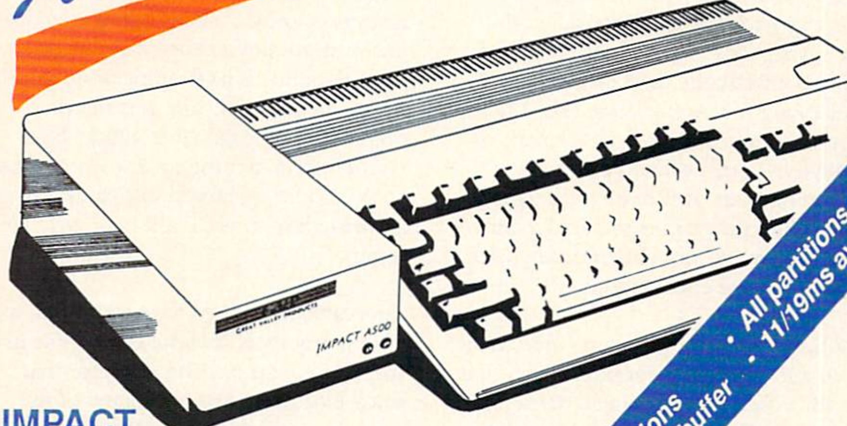
Dear AC:

For those bothered by the interlace flicker, and the price of flicker fixing boards (availible only for the 2000), the Flicker Master™ is a low cost fixer. The Flicker Master™ is a screen that attaches to the front of your monitor via velcro. This screen improves contrast, reduces eye strain, etc. in addition to greatly reducing flicker. The Flicker Master™ fits the following monitors: A-1084, A-1080,

AMIGA 500 USERS! NOW GET A2000 POWER!!

Announcing

THE WORLD'S FIRST AUTOBOOTING HARD DISK FOR THE AMIGA 500



- AUTOBOOT directly from hard disk
- Internal & external SCSI connections
- All partitions can be FFS with V1.3 AUTOBOOT
- Connect up to 6 SCSI peripherals
- DMA data transfer to onboard 16K buffer
- 11/19ms average access time on 40Q & 80Q models

IMPACT A500 – HD/RAM SUBSYSTEM

offers everything any A500 owner has ever dreamed of – in one compact, easy-to-install, add-on subsystem; provides the same high performance as our top-of-the-line A2000 controller

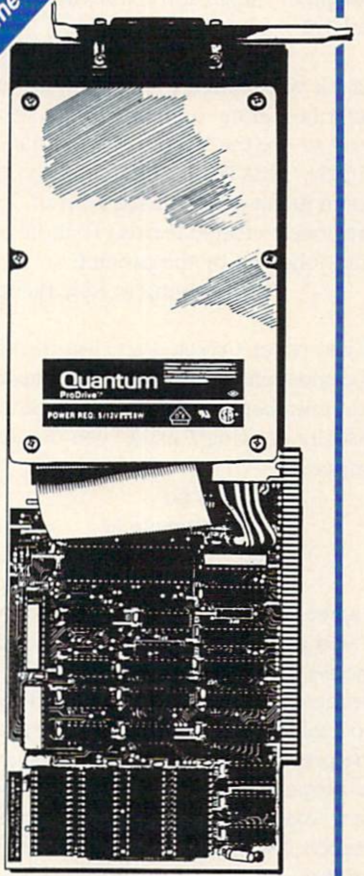
IMPACT™ A500 – HD/20
A500 – HD/30
A500 – HD/40Q
A500 – HD/45
A500 – HD/80Q

Available with a built-in 20, 30, 40, 45 or 80 Mega Byte Hard Drive and an optional 2MB plug-in FAST RAM/AUTOBOOT expansion module.

A2000 USERS!

Still

THE MOST POWERFUL HARD DISK SOLUTION FOR YOUR A2000!



IMPACT™ A2000 – Hard Card 30
A2000 – Hard Card 45
A2000 – Hard Card 40Q
A2000 – Hard Card 80Q
A2000 – Hard Card 100

The GVP IMPACT-HC series of "Hard-Cards" for the Amiga 2000 matched with the 3.5" Quantum ProDrive hard disk drives, provide turbo performance and unbeatable value at list prices as low as $18.00 per Mega Byte! The Quantum ProDrives provide average access times as low as 11ms with their built-in 64KB read-ahead cache and offer a MTBF of 50,000 hours which is almost twice the industry average.

Now Shipping **AUTOBOOT EPROMS**

GVP

GREAT VALLEY PRODUCTS INC.
PLANK AVENUE, PAOLI, PA 19301 TEL (215) 889-9411 FAX (215) 889-9416

Sony KV-1311, Magnavox RGB 80, NEC Multisync, and others. The screen sells for $16.95. The manufacturer is: T.S.R. Hutchinson Co., 110 W Arrowdale, Houston, TX 77037-3801, Phone (713) 448-6143.

Sincerely,
David N. Witmer
Stevens, PA.

Dear AC,

Paul is responding to a letter from a reader who was having trouble with the Computer Aided Instruction, article in AC Vol 3.9.

Thank you Jimmie for correctly pointing out a bug in my program. When you want to use the Editor, and you choose "OPEN/CREATE LESSON" from the pull down menu, you will see a list of previously created lessons (if there are any) followed by the prompt:

```
Enter existing or new filename:
```

If you haven't created any lessons yet, you must enter a filename. However, if you mistakenly hit return, without actually entering a name, you do get the response:

```
There is no
Want to create it?
YES          NO
```

I agree that this is a silly response. Also, if you pick YES at this point (with left mouse button), the program seems to become stuck. But it really isn't, for if you select NO, the program returns to the previous prompt, thus allowing you to properly enter a filename for your new lesson. If you then enter a name, French_1 for example, you will see the prompt:

```
There is no French_1
Want to create it?
YES             NO
```

And now, if you select YES the program correctly creates your new lesson. I realize that this part of the program is a little confusing and needs cleaning up in the next revision.

Remember, for the program to operate properly you must install on one floppy disk the following files:

AmigaBASIC
Tutor
Editor
diskfont.bmap
graphics.bmap
empty directories: Lessons
This Disk

I intentionally wrote the program to require this because it simplifies operation for users of single drive systems. There is enough room left on one floppy disk for over 1000 questions!

I'm glad you finally got the program to work and I will certainly consider all your suggestions for the next revision. Your's is exactly the kind of feedback I was looking for. At the same time I hope that this program lights a small spark of initiative in you and that you consider performing your own modifications to it. After all, the program is public domain. Anyone is free to do whatever they like with it, in fact, I encourage it.

I also hope that this program, and others like it, are interesting enough to you that you will consider becoming a subscriber to Amazing Computing. I am busy right now writing a series on fractals and I wouldn't want you to miss out on them.

Yours Truly
Paul Castonguay

If anyone is having trouble with a program appearing in AC, please write the author c/o of Amazing Computing. Although most of the writers for AC are freelance and not staff, all are Amiga fanatics.

We will forward your letters and concerns to the appropriate authors, who, if at all possible, will respond courteously and quickly to their fellow Amiga enthusiasts!

Dear AC

We read with surprise and concern Amazing Computing's V3.8 "Roomers" column, in which a lawsuit involving Electronic Arts, Bethesda Softworks, and MaxiSoft were addressed. Although you print the message at the end of the column that "The Bandito" cannot be held responsible for the reports made in the column, I doubt you would disagree that the sign of a professional journalist is one who takes responsibility for checking his facts and for the slant with which he writes his column.

Had "The Bandito" checked his facts on the information he reported he would have learned that Bethesda's fraud, unfair business practices and bad faith charges have been dismissed by the court. What remains is a dispute regarding the terms of the contracts between parties. Further, the amount claimed by Bethesda is now $400,000-not $7.5 million. In addition, EA is confident that Bethesda's contract claims will ultimately be dismissed as well. Finally, EA's lawyers had filed EA's motion to dismiss those charges weeks before your column appeared. These are matters of public record and should be checked out before they are published. "The Bandito" is misleading your readers. As for the Maxisoft suit, it is in our opinion an unreasonable attack. Had your editor taken the time to contact us, he would have learned that, thus allowing him to write a factual and fair report.

Anyone has the right to sue another, at almost any time. It should not come as a surprise to you or "The Bandito" that since Electronic Arts and many of its products are critically acclaimed, we are an easy target for random lawsuits.

It is our opinion that the "The Bandito" has attacked Electronic Arts' credibility without cause. I hope you will publish this reply, and thereby correct the misperceptions created by the column.

Very Sincerely,
Randall Thier
Vice President, Affiliated Labels

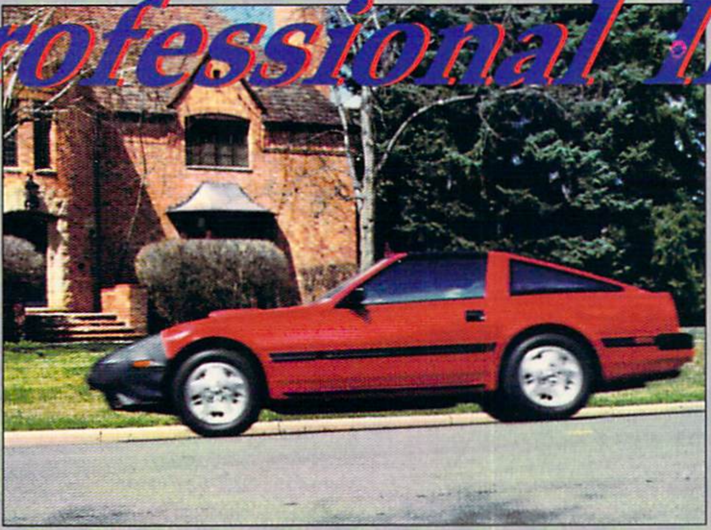
We apologize for any inconsistencies between the statements in the "Roomers" column and the facts as they were at the time you read them. However, you are certainly aware of the time difference between when a story goes to press in a magazine and when it is seen on the street. Most stories are written two to three months before they appear in print and since nothing stands still, this does lead to inconsistencies.

The story, as it was published, was directly from releases by reliable parties involved in the disagreement. The depth and detail of the information, left little room for misunderstanding. The suit was filed as stated.

Do you have an Amiga related concern?
Please Write!
Amazing Computing Letters
P.O. Box 869
Fall River, MA 02720

PROGRESSIVE
PERIPHERALS
& SOFTWARE

The Professional Image!



Photography M.T.Morrissey

ProGEN
GENLOCK

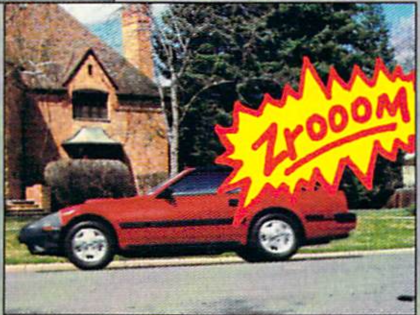
FRAMEGRABBER™
Real Time Video Image Digitizer

Captures 4096 colors in 1/30th of a second

ProGEN is a must for Amiga desktop video! Editing studios, artists, television studios, in-house production departments can all use **ProGEN** with their Amiga 500, 1000 and 2000

- Make your own desktop video productions
- Overlay Amiga Graphics on any video signal
- Transfer images from your Amiga to VCR

- Software-Selectable Foreground, Background, Amiga Out and Video In
- Select one of 32 colors as transparent for video effects
- Now available from Progressive Peripherals & Software, Inc. or your local dealer
- Meets RS-170A standards



FrameGrabber Real-time Video Image Digitizer for the Amiga 500, 1000 and 2000

- Grabs images from VCR or home video camera
- Captures color images in 1/30 of a second (1/60 of a second for black & white)
- Supports screen resolutions from 320 x 200 to 640 x 400 Also supports overscan (352 x 240)

- Includes image processing software from the creator of PIXmate!
- Automatic Time Lapse Animation feature
- Multiple exposure mode
- Stores images in IFF format or as raw RGB images
- Allows you full control of Hue, Color and Saturation with conveniently placed control knobs

Progressive Peripherals & Software, Inc. • 464 Kalamath St. • Denver, CO 80204 • (303) 825-4144

Amiga 500, 1000 and 2000 are trademarks of Commodore-Amiga, Inc., ProGEN, FRAMEGRABBER and PIXmate are trademarks of Progressive Peripherals & Software, Inc.

How may I animate Thee? Let me count the Ways-

by R. Shamms Mortier, PhD(c)

The wonderous images that wind their way across the Amiga screen did not suddenly arrive with the introduction of Commodore computers, or even with computers in general. Animation began even before film itself entered the picture, with the "magic lanterns" or zoetrope wheels. Even before that, painters would now and again try to elicit motion by doing a series of static paintings and placing them side by side. I remember seeing a series of images that the painter Goya produced in the early Nineteenth century. As I remember, the images depicted a monk being robbed, and they were placed in succession so that the eye was forced to take them in one after the other. They were static, but a definite movement was implied. And who has not been exposed to the futurist "Nude Descending a Staircase", an experimental painting that tried to show where the image had been, as well as its intended future direction. Because movement is commensurate with life itself, the urge to depict static form in a way that implies movement has been the visual artist's obsession for millenia. In the Paleolithic caves, splendid red ochre wall paintings show Bison and Antelope with six and eight legs,..the work of some pre-dawn-of-history protoanimators.

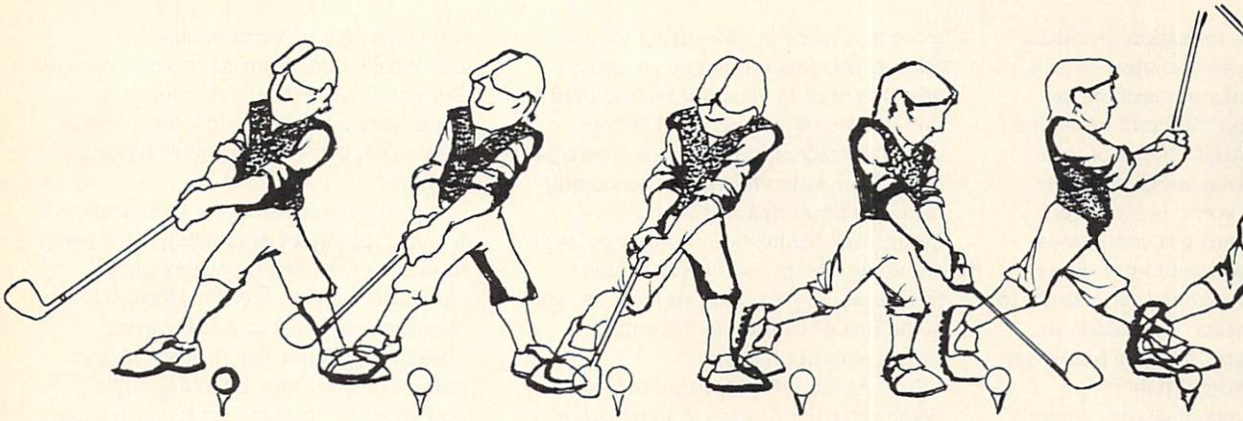
Certainly animators existed before Disney. But he had one talent the rest of them lacked, and that was the ability to gather working groups of very talented individuals in one place in order to focus all of their energies on singular projects. It is with Disney that the main child of the Industrial Revolution, assembly line production technique, is allowed to jump into bed with the arts. From that marriage in 1928, the Rodentia Magnificat, Mickey Mouse, was given birth. Right up until this very day, from the political adult animations to the Saturday Morning Mania, the art of animation continues to be the result of a many faceted collaboration of large numbers of individuals on each production. But there's some revolutionary changes a-comin'.

The main lure of computer graphics, especially that corner of the field flowering forth from developers dedicated to the art of the microcomputer, is that the individual, the lone creator rather than the group, can regain the ascendancy as the all-around animator. Yes. Large projects with short deadlines still work better if the work is spread among several animators, each dedicated to a specific task or characters. But if the deadline is only one of convenience (or if it is realistic enough) then it is completely possible for a single individual to do all of the animation on a computer. By "all of the animation" I mean the original concept, the storyboarding and layout, the drawing and colorizing, the titles, credits...AND the soundtrack. Aside from being an egoistic enterprise (and still a very long and dedicated process) the mere fact that it is possible (with very professional results) breathes new life into a traditional industry.

In my opinion, the animator who would stretch her/his wings in the Amiga nest would still be best served by studying the variety of traditional methods and terms used to produce animated visuals. Why? Well for one thing, the guild demands it. Respect comes in part from dues paid. But there's more than that, more than the love that suffering has for company. There is also the fact that certain techniques and methods are best learned slowly over time with a master of the craft. Art is more than the production of a thing, it is the involvement in a transformative process. That very process gives back, releasing options into the blood stream of the individual over time.

Aside from the philosophical and aesthetic considerations is the exposure of the animator to a variety of techniques. Many of the Amiga animation packages to recently hit the market seem to dwell upon only one way of producing meaningful animations, as if the art itself were boxed into a definatory style and direction, usually equated with the work of Disney. How many recent packages have you seen that compare themselves with "flipbook" style productions? I've seen quite a few. It might be best, then, to share with you a number of animation methods that are also doable on a microcomputer, and any of which (or perhaps several in combination) can be used to produce "meaningful" visual pieces.

Let's start with the flipbook method, also called "cel" animation. Traditionally, cels are sandwiched pieces of acetate painted with special polymer paints. Each cel is meant to be joined with a background, which can be seen because there are areas of the foreground cel that are left unpainted and transparent. When the two (or three, four...) are joined together, they are usually placed on a copy stand, and the entire scene is captured with a camera on a single frame (or two or three) of film. Most of the advertisements that accompany Amiga animation software, attempt to center upon the cel animation process. For the beginning or novice computer artist, it sometimes seems as if mastering this procedure alone would



qualify one for admission to the animators arcane guild. Not so. There are several other ways to shape an animated piece, and all are doable in some fashion on the Amiga.

Now in addition to the software that I'm going to mention, there is a standard list of hardware items that you should consider procuring (either through single purchase or shared expense with other animators) that will allow you to develop your finished work in a most professional manner suitable for showing and broadcast. As far as the hardware is concerned, it is not inexpensive, and the vendors (and quality) vary. This is an area so far neglected by Amiga or microcomputer third parties. All that exist here are the sources that have answered professionals over the years, and some of the vendors have bloated their prices accordingly. Just as we have a wish list for economical products to answer the blossoming desktop publishing needs of home businesses (real-world priced laser printers, etc.), there is a growing need for qualitative yet lower priced animation tools. Perhaps even big C itself should be thinking of entering this marketplace. For qualitative results the computer/video animator needs a professional video deck (with "flying erase-heads"), a single frame controller, maximized storage capability (a hard drive, preferably with removable/replacable media, or a read/write CD ROM device), access to editing decks and controllers, and at least two megs of RAM on the system. All of this in addition to the computer hardware and software. Oh yes. If you're going to be producing images that are generated in 3D on your screen, then you should also think about an accelerator board with a 68020 or 68030 processor, and a math co-processor chip. In animation, time is not only money, it is also sanity. Digitizers, Frame-Grabbers, and Genlocks are also valuable here.

So, what other animation techniques exist besides cel animation? Let's start with something called "process animation". Process animation allows you to witness the evolving artistic process as an integral part of the "feel" of the story. We've all witnessed this technique at one time or another, and whole beautiful pieces have been created by just utilizing this method alone. In process animation, you may see a figure being drawn before your eyes. After it is outlined, colors magically fill areas of the work. All of the while, the narration is superimposed over the visuals, carrying on with the story. Anyone with a video machine and a paint package could at least begin to experiment with process animation. DPaintII, PhotonPaint, ExpressPaint, Digi-Paint...all will work. Best to plan your moves first, maybe even tape a tracing paper version of your work to the screen as a guide. I prefer to do this without the obtrusive cursor, so I disappear it. This also adds to the magic. The variety with this technique alone is limitless.

Of course there's 3D computer generated animation. Here's where you really need a single frame controller and a top of the line recorder that it can address. Usually the 3D packages (Caligari, Sculpt 4D, Animation Apprentice, Forms In Flight II, and Turbo Silver, to name a few) have dedicated drivers for a single frame controller, or will allow you to address third party hardware. The idea here is to develop only the "key frames", the points in your story that are major orientations. Later, the computer can grind away and fill in the "in-betweens". It's best here if you also have an accelerator board installed to speed up the process. Animations of this type really require post production editing of sequences, so access to that level of hardware is vital. Before you can engage in this animation, you have to design and create the 3D objects and elements...which is where Aegis Modeler, 3Demon, and other dedicated modules come in handy. With the Syndesis handshaking programs, it is possible to design with one package and animate with another. Added to these would be the animation utilities, of which the Amiga can boast a superlative line that keeps getting bigger: VideoEFX 3D, Photon Cel Animator, PageFlipper=FX, DVideo 2.0, and DeluxeProductions come to mind, readily.

"Tracing Paper Animation" is yet another "look" or form. With this procedure, the aim is to allow the previous frame to "show through" the one you are shooting. The final result is a piece that allows the moving elements to leave "traces" as they fly around the screen. My teacher and master, David Ehrlich (a world class animator) uses this method exclusively and with hand tools. There are only a few packages that allow this on the Amiga at present. The "Pencil Test Mode" of Photon Cel Animator, and the "Strobe" modes of other packages. Perhaps someone will take the hint and expand upon this capacity. Of course another way to accomplish Tracing Paper animations is to allow your digitizer to do the work. Do your separate frames on tracing paper, and digitize either two or several while leaving the previous ones showing through the present frame. I haven't done it yet, but I'm betting that there's some interesting results possible.

(continued)

Torn Paper animation produces some wonderful and evocative results. This method is really a cousin of the process method, but different enough so it is in a class by itself. Pieces of torn paper or fabric move under and over each other until a scene is produced. Narration and/or music is continuous. With the Amiga, you could use either paper or other 2D material (ie., fabric) to achieve similar results. In addition to digitizing your work frame by frame, you could create electronic "paper" composed of polygonal shapes, entirely on the Amiga screen. Perfect programs for this would be Aegis Animator, Broderbund's Fantavision, and Zoetrope from Antic Software. You could also digitize elements from newspapers or books for that "Monty Python" look. Canadian animators, recognized the world over as some of the most experimental and creative around, have pioneered in using shifting shapes in sand as variable form. With the right Amiga equipment, this method would be open to you as well.

"Object Animation" exists in a world all of its own. One piece that I did some years back (with super 8 film) was to allow my keychain to chase my key around a table for about thirty seconds. It was quite effective. Any 3D objects would do, as movement brings them to life. With the Amiga and a single frame controller (or many individually digitized frames), and a video camera pointed at a work surface, you could do some very rewarding experiments in the "Object Animation" category.

Pixelation can be defined as allowing some moving footage to become vagued out and smeared by purposefully allowing the small picture elements of the screen (Pixels) to become enlarged. On static images, "Butcher" and "Pixmate" give you many ways to experiment with this. This technique is used much in music videos. With the A-Squared "LIVE" hardware (and also the software called "InVision from Elan Design), you can pixelate moving images till you drop (or your audience does). With the addition of "Mandala", you can also superimpose live actors into the scene, much as was done in Disney's most endearing features.

"Rotoscoping" is the procedure whereby live footage is translated into painterly images. This is being used more and more in advertising on the tube. It was first pioneered by (guess who?) Disney in Sleeping Beauty back in the Thirties. With a series of frames digitally resident in the Amiga (perhaps taken from video footage) rotoscoping becomes an Amiga reality. Hash Enterprises "Animation: Rotoscope" will allow the Amiga user to utilize this technique on personal productions, and is the first of its kind in the entire microcomputer industry.

As far as I know, nobody has yet dedicated their Amigas to Claymation productions. Will Vinton's California Raisins still make their way to your screen using traditional Claymation techniques. But it is possible. Claymation takes a lot of time and a lot of planning. Multiple figures have to be created to replace those that melt under the hot lights. Any of you Amigans interested in the adventure?

Don't you love those Spielbergian clouds that flutter across the sky? Well, the animation technique that is used to produce many of them and other effects as well is called Stop Motion Photography. Through stop motion, we can see an apple slowly give way to the pierce of an arrow, and we can witness the sacred hidden magic of the opening Rose to the light. Now, with the Transport Controller from MicroIllusions (and the other necessary hardware) we can produce Stop Motion effects with the Amiga.

You can construct and move a multitude of puppets and process the results through the Amiga. They can be constructed from a wide variety of materials in either two or three dimensional versions. Puppetry is one of the oldest forms of animation. You can be among the first to translate it to the Amiga screen. I remember reading that the creator of Saturday Night Live's "Mr. Bill" was experimenting with the Amiga.

Many industrial and instructional animations begin and end with "Flow Charting". The Amiga has the capacity to produce awesome flow charts. The colors can either cycle (DPaintII) and/or can shiver and reform as morphed polygons (Aegis Animator, Fantavision, and Zoetrope). It is amazing what a clear narration and some flowcharting can bring to consciousness. I have produced literally hundreds of pieces on the Amiga utilizing color cycling and polygonal metamorphing alone for instruction at the University of Vermont and for industrial clients from my home business. Both are unique forms of animation made very available by the advances in microcomputer (and specifically Amiga) graphics.

What you strive for in animation is a "look", a way of organizing the moving visuals so that they themselves tell a story and come to life. This look can be as representational as Disney or as abstract as a latter day Picasso. "Looks" carried on over time becomes "style", and style becomes the artists trademark. Style is achieved over a long period of time by risk and experimentation, by trying to bend and warp the processes in unexpected ways. Even the experience and history of the animator cannot supercede the taking of necessary risks to achieve welcome surprises. Risking and stretching into the realm of the unexpected is what an animator does constantly, and in the realm of computer graphics and computer enhanced animation, soon we will all be able to say with confidence..."Amiga does it better".

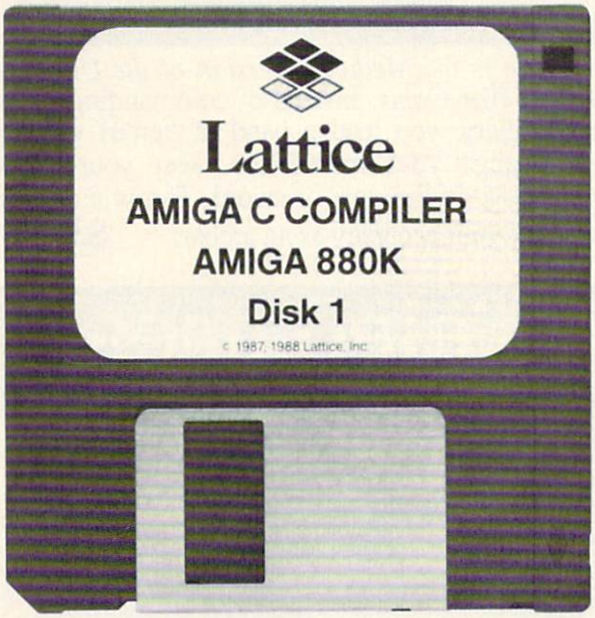
•AC•

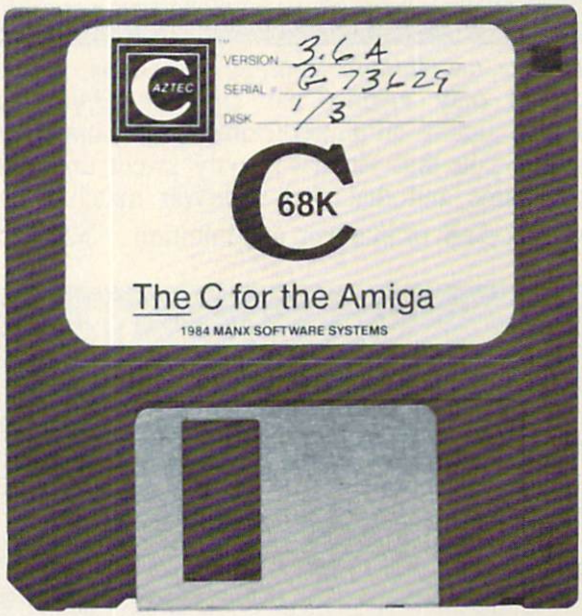
Biographical Notes

R. Shamms Mortier is a graphic designer with some thirty-five years of experience in all areas of visual design and production, including soundtrack composition for film. His work has been presented in professional galleries. The production of his animated film "the Secret Dreams of Older Men" was funded by the American Film Institute/Boston Film-Video Foundation, and he hopes to translate and revise/enhance it in the future with the Amiga. He has prepared instructional demonstration tapes for industry, including one based upon atomic orbitals for secondary school learners (MacMillan and Co.) utilizing Amiga graphics. His doctoral studies (Union Graduate School, Cincinnati) focus upon symbolism and upon the transformative possibilities of the arts. Currently, he is the manager of the Graphics Service of the University of Vermont, and owns and operates his own business (Eyeful Tower Communications) in Bristol, Vermont. He teaches graphic design at the UVM, and art/transpersonal psychology courses at Burlington College, Burlington,Vt. He is most open to comments and criticisms on articles and reviews, and can be reached via this magazine.

VROOM VROOM.

SPUTTER SPUTTER.





Introducing new Lattice C for Amiga DOS, Version 5.0.

In recent benchmarks, Lattice C 5.0 was off to the races before Manx 3.6 was even off the starting line. In fact, statistics prove that no other language, no other compiler, gives you faster, more efficient programs. Period. And it now supports 68020 and 68881.

BYTE* Benchmark	Lattice C 5.0	Manx C 3.6	% Difference
dhampstones	42.4	62.7	32%
dhrystones/sec.	1605	1017	58%
matrix manipulation	15.7	50.4	68%
scbigrd	70.2	82.8	15%
sorto	76.4	110.9	31%
sieve	34.4	40.5	15%

And it's the most comprehensive, exclamation point. Nothing else gives you such a complete programming environment. All these tools and utilities are included in the Lattice C 5.0 package:

- Lattice C Compiler • Compiler Companion Utilities
- Source Level Dubugger (CodeProbe) • Code Profiler
- Global Optimizer • 304 C Function Libraries • Blink Overlay Linker • Assembler • Disassembler • Librarian
- Sample Files & Examples • Lattice Screen Editor • Installation Program • Comprehensive Documentation.

Our new, easy-to-use, Source Level Dubugger, *CodeProbe*, supports both C and Assembly language, and multi-tasking debugging. And our new Global Optimizer will actually *enhance your programs' performance up to 40%.*

So choose Lattice if you want to program on the fast track. Or choose Manx and finish at the back of the pack.

To purchase Lattice C 5.0 or for more information, see your favorite software supplier.

Lattice

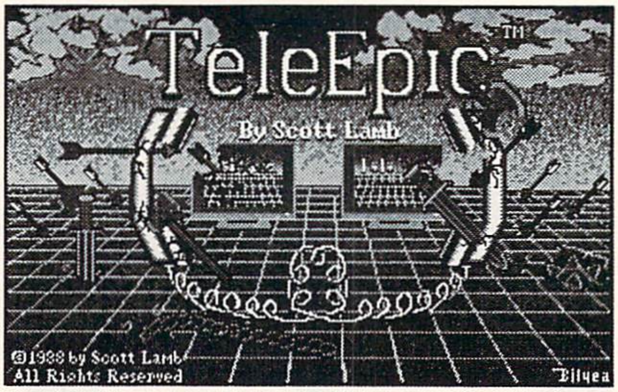
Subsidiary of SAS Institute Inc.

Lattice, Incorporated
2500 S. Highland Avenue
Lombard, IL 60148
Phone: 800/533-3577
In Illinois: 312/916-1600

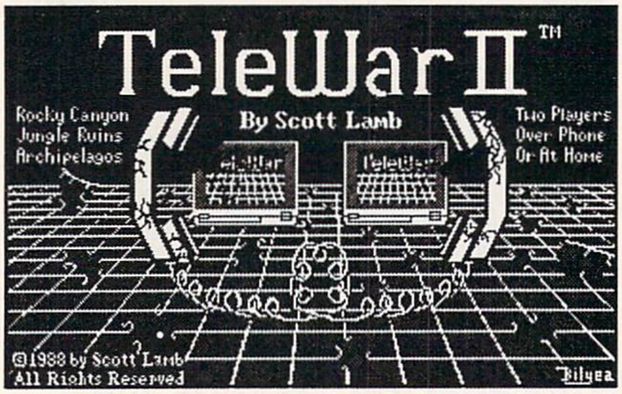
*BYTE is a registered trademark of McGraw-Hill

THE ULTIMATE STRATEGY BATTLE SIMULATIONS

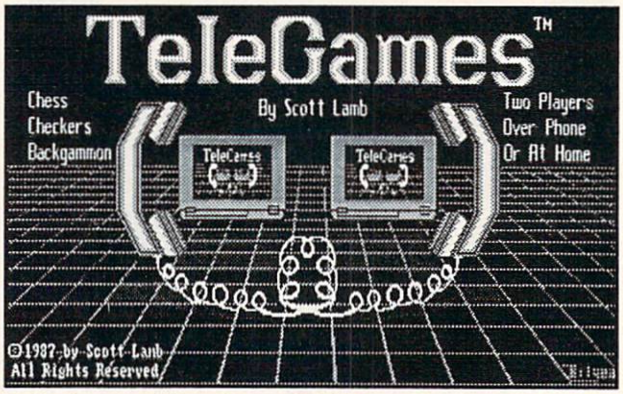
TeleWar, the *Ultimate Strategy War Game*, allows you to play against the *Ultimate Opponent*. Playing against an unpredictable human opponent assures you that victory is only sweet until the next game and that each TeleWar match is an exciting clash of intellect and intuition. **$39.95**



TeleEpic is the *Medieval Version of the Ultimate Srategy War Game*. Instead of commanding tanks and artillery, you lead a band of heroes waging war against your mortal foe. Test your skills wielding a mighty axe or sword. Throw a dagger or check your accuracy as an archer. **$39.95**



TeleWar II, the *Ultimate Strategy War Game Revisited* with new terrains and enhanced game play. Experience the ecstasy of blasting your best friend to rubble or disintegrate someone you don't even know. Watch out! You may be the next casualty peering out of a twisted pile of smoldering steel. **$39.95**



TeleGames , like all our '*Tele*' products, provides the enjoyment of competition without the problem of where to play. Competition is enhanced with smooth 3D perspective graphic animation and digitized sound effects. Players may compete on one computer or 'TeleConnect' via modem. **$34.95**

INTRUDER ALERT

The ***Intruder Alert Monitor*** allows your computer to become a monitoring, logging, or security system for your home or business. The system includes a startup kit including several types of detector switches and wire. Each of eight individual ports may be programed independently. Intruder Alert software allows for digitized audio alarms, speech synthesis, modem interaction, date and time stamping of events, external program launching, and utility support for BSR/X-10 controlling. Like all Software Terminal products direct support from the author and designer is available though our BBS service or via BIX vendor support area 'TeleGames'. **$159.00**

AVAILABLE FROM YOUR LOCAL DEALER!

SOFTWARE TERMINAL

3014 Alta Mere Drive
Fort Worth, Tx 76116
Phone: (817)-244-4150
BBS: (817)-244-4151

Alive in 3D...

A review of Caligari

A high-end 3D sculpting and animation package

by R. Shamms Mortier

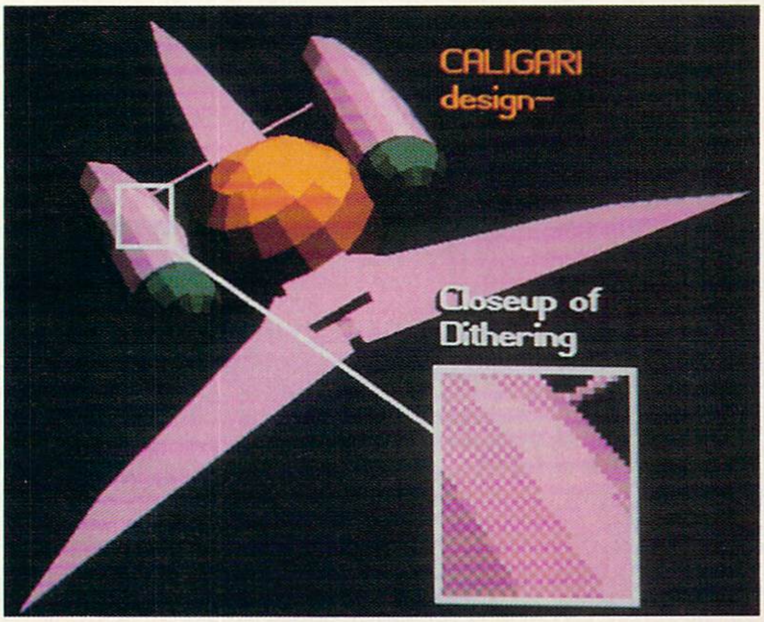
Background

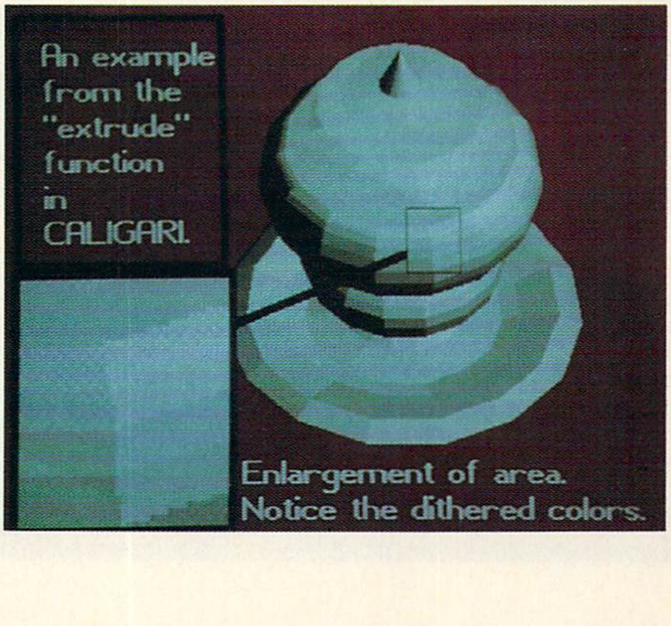
When I began using the microcomputer to produce visuals around seven years ago, what was referred to as "state-of-the-art" was a far cry from our minimal expectations today. I remember about ten of us crowding around an Apple screen, watching the lines we were tracing on an ancestral digitizing tablet replicate themselves on the screen. Of course, we had fantasies of being able to utilize more than sixteen (or was it eight) colors, of manipulating objects in a 3D environment, and of translating various realms into visual form. At that time, the Commodore-64 was just being introduced to the market, and upon securing the necessary funds from the credit union, my present obsession was firmly rooted in an exploratory reality.

Of course, other recourses were available even then. With a little stiffer loan (around thirty to a hundred thousand dollars) I could have opted to tap into one of the vendored graphic work stations being offered. But I just couldn't see doing that, so I had to content myself with packages like Sorcerer's Apprentice and Flexidraw on the C-64 while Cranston-Cusiri productions was designing all of those magical graphics that fluttered around the screen during NFL halftime breaks. Little did I suspect the graphics revolution that was rushing my way.

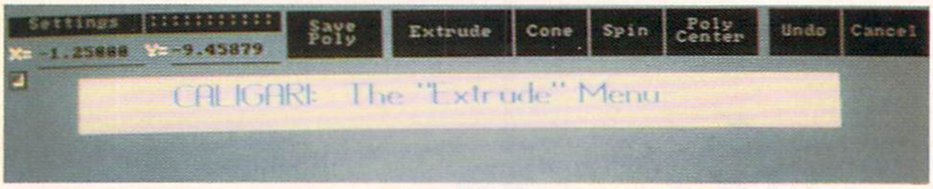
I have been a graphics designer and producer for over twenty years, and have attained a moderate degree of proficiency in a number of mediums, from illustration through page preparation. Having experimented with it all, I am not easily impressed by new graphics tools, although I maintain an openness to any process or technology that will allow me to produce qualitative visuals with ease and variability. That is why I chose the Amiga in the first place, and it is also why I pleaded with the University of Vermont (where I now manage the Graphics Service, designing and producing instructional visuals) to invest in it.

Over the last two years, my point choice has been validated. I have managed to purchase software that equals what would have cost hundreds of thousands of dollars worth of capability just a few short years ago, and the development is just beginning. You name the graphics package or the hardware interface, and I've either got it or am making devious plans as to its procurment. This is a path of no return,

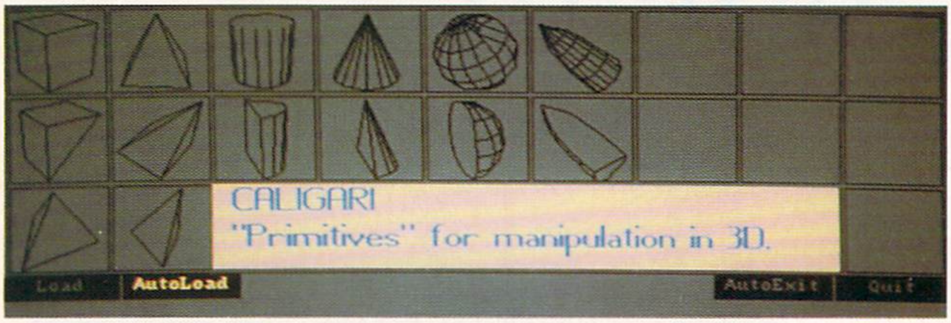




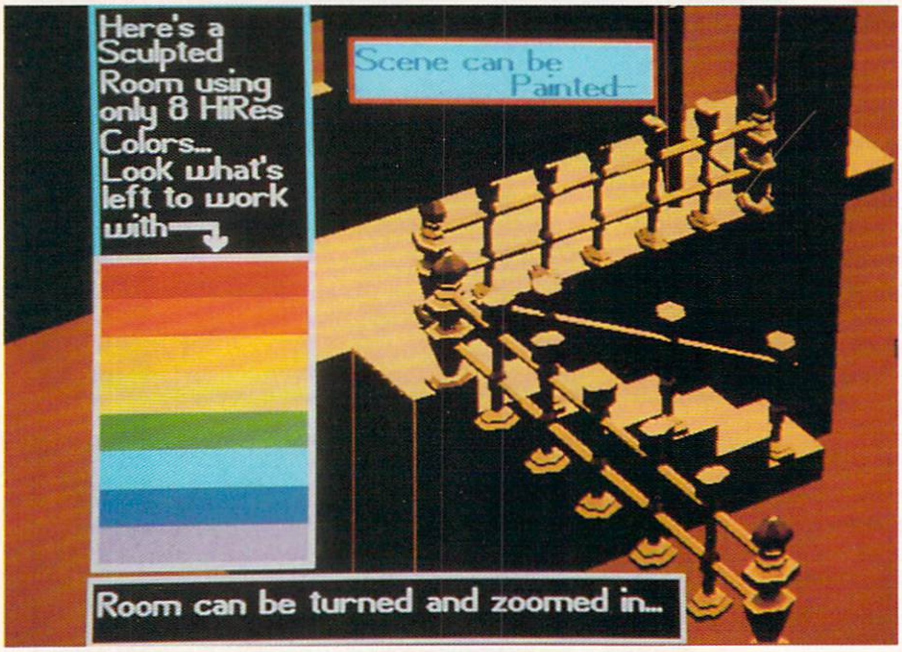
(continued)



CALIGARI: The "Extrude" Menu



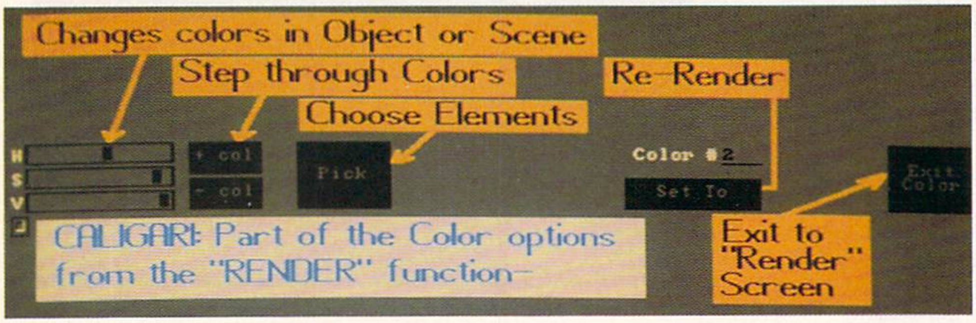
CALIGARI
"Primitives" for manipulation in 3D.



Here's a Sculpted Room using only 8 HiRes Colors... Look what's left to work with

Scene can be Painted–

Room can be turned and zoomed in...



CALIGARI: Part of the Color options from the "RENDER" function–

not that I'm having any second thoughts in anyway. I've also noticed that my business has gone through a radical shift in the last two years, comensurate with the development of the Amiga. I am being called upon to produce more and more slide and video graphics, while the standard processes (those aided and abetted by the "tee-square" and the triangle) slowly atrophy into the distance. Graphics Design is slowly becoming Computer Graphics Design, and new tools and techniques are in constant demand.

Money

When I first began my Amiga obsession, the high-end cost of a software package was around two hundred dollars. We all used to gloat over the discrepancy between Amiga software costs and the cost (and quality/variability) of IBM packages (not that many of those packages could do what the Amiga packages could). Of course, there was only a limited amount of software availability for the Amiga then (Remember?!!). Most of the software that I use everyday costs less then a hundred dollars a package, and some of the most useful much less. So why even consider speaking about a two thousand dollar package (and that is the "pre-release" price!)? Well, it all depends upon the level of your involvment.

Dialogue with a developer

I dislike writing cold reviews of products, and instead prefer to pepper my comments with reflections gathered either from my own experience and/or (best case scenario) remarks and anecdotes drawn from direct conversations with the developers themselves. I've always found the most creative people are the most generous with their time in granting such interviews. Roman Ormandy, the originator of Caligari, fits this pattern.

Many of us have known about the development of Caligari for quite a while, ever since its speculative debut at SIGGRAPH some years back. At that time, Octree presented a videotape that allowed the viewer to travel between and amongst a graphic cityscape of looming shapes and byways. The presentation of this video, and rumors concerning the direction of Amiga software development just around the corner, sent expectatory shock waves

(continued)

zo·ë·trope (zō´ə trōp)

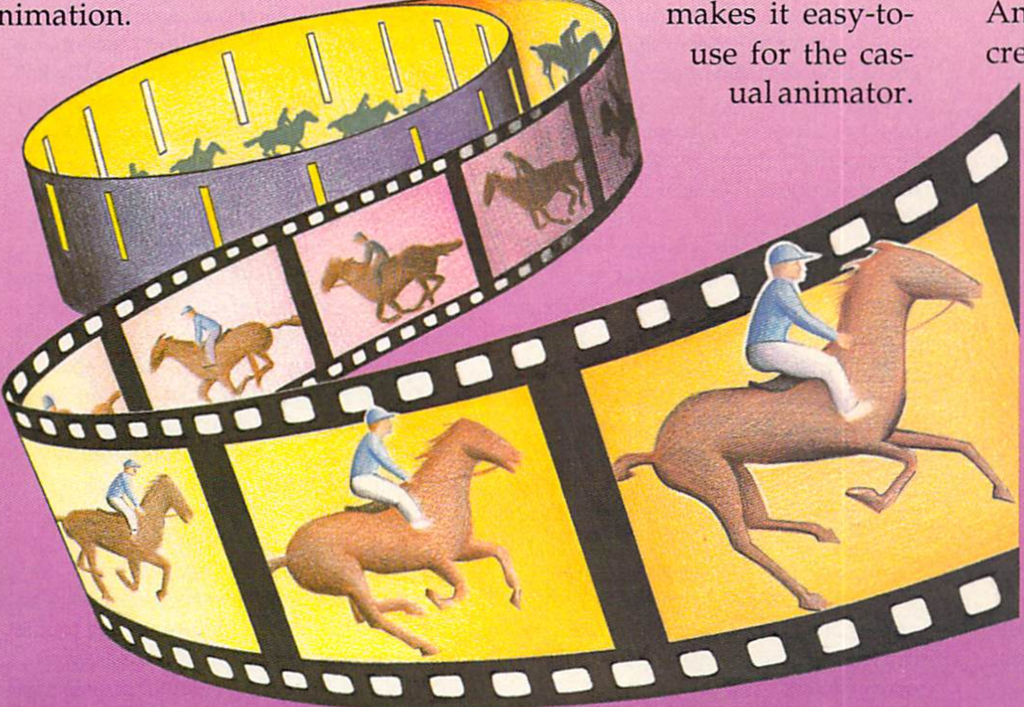
n. [irreg. f. Gr. *zoe-*, life + *tropos*, turning] 1. a device that gave static images an illusion of motion. Known as the "wheel of life", the zoetrope brought the magic of animation to the parlours of the mid-1800's. **2. a new state-of-the-art advanced animation system for the Amiga® computer from Antic Software.**

Now! The program you've been hearing so much about... Zoetrope puts you right on the cutting edge of animation.

Its high end features satisfy even the most demanding of professional users. Its unique layered structure makes it easy-to-use for the casual animator.

Bring your imagination to life with Zoetrope: the faster way to animate.

Now with Zoetrope, The Animation System™, *you* can create magical motion.



Here are just a few of Zoetrope's exciting, advanced features:

- Spin and swoop titles and images through 3D space.
- Apply computer power to classic "cel" animation.
- Overlay and underlay (compositing) your favorite pictures or animations.

Taking animation into the 21st Century

By Jim Kent, the creator of Aegis Animator*

ZOETROPE
THE ANIMATION SYSTEM™

Another great Cyber Graphics product.

Antic Software

To see the entire line of Cyber Graphics from Antic Software, visit your favorite Amiga® retailer.
To order direct, call 800-234-7001.

MM8AW3

Amiga® is a registered trademark of Commodore-Amiga, Inc. Aegis Animator is a trademark of Aegis Development.
Zoetrope, The Animation System; Cyber Graphics; Antic Software and Antic are trademarks of Antic Publishing, Inc.

from one end of Amigaland to the other. No doubt, this videotape either enhanced the developments of other developers, or may have catalyzed some developers into particapatory action.(This piece of videography, by the way, is included on an instructional videotape that accompanies the release cum pre-release of the software). When I read about that presentation, I immediately found Octree's phone number, and begged Roman for more information. He was most congenial, and explained the actual release was a way down the road, but said that he would take my name and keep me informed. I shuddered at the prospective investment, and filed the conversation away in a fantasy drawer.

It was with a stir of memory that I opened a letter from Octree in September of 1988 listing the availability of the pre-release version, including a price sheet of addendum hardware suggested for its use. Again, I reached for the phone. There's been too much vaporware on the market lately, too many vendors offering their wares for sale and then using the money to continue development. I dislike waiting six or more months for something advertised as "available now", so it's become a habit to call for the details before I sign checks or give out credit card numbers.

Speaking with Roman Ormandy can be like having a conversation with an electronic Picasso or a mensch in any other field of endeavor, if you ask the appropriate questions. Caligari, I learned, was not only at the introductory stages of its planned development with many upgrades promised to owners, but the sheer depth and breadth of Mr. Ormandy's knowledge and interests may cause future seismic eruptions in the visual dataverse. I was certainly left with the impression that I had spoken to someone whose mind roamed (as in Romin') with ease in fields as seemingly diverse as philosophy, psychology, art, and AI.

Roman Ormandy is an emigree from Czechoslovakia, in his own words a defector. Some of us may think stories of America as a continuing land of freedom and opportunity are either passe' or chauvenistic, but when you meet someone who has been able to struggle and succeed in an historical American fashion, a certain kind of shiver runs up your spine, even if you keep it secret from your more jaded friends. He has an advanced degree in Artificial Intelligence from the Prague Department of Computational Linguistics, one of the more advanced and prestigious institutions in Eastern Europe. He has also studied computer sciences and psychology, and mentions exposure to the work "A Philosophy of Symbolic Forms" as a seminal text in his development. After saving enough money to fly to SIGGRAPH in 1985, he was exposed to and fell in love with graphics and visualization on the computer. Bob Periseau made a presentation with two Amigas at that meeting which immediately "turned Roman on". He was not a developer at that time, but was concerned about the issue of the "human interface". He wrote the prototype for Caligari in the next three months.

Realizing the prototype needed the kind of finnese only a master programmer could give it, he formed an association with Peter Kennard, a programmer with very strong design experience who reprogrammed the module from scratch. Mr. Ormandy insists that the two and a half years that Peter spent working on Caligari was the main reason it was implemented, and he praises Peter's work unashamedly.

You may not agree with his philosophy in toto, but you have to appreciate the depth of his thought process once you engage him in conversation...

"Technology defines the meaning of human life, even though we might be much more human without it. The strongest potential of 3D graphics communication is that it can enhance all of human communication. The original idea of AI was to create a human alter ego, which could enhance human capabilities. AI is the strongest science, and it can bring an expansion of human capabilities. New models will deepen our concepts of evil and goodness."

On November the 22nd, 1988, Roman Ormandy will engage in the legal ritual that will make him a citizen of the United States. He complains that this is too late for him to be able to vote.

The Software

Caligari comes with an oversized manual packaged in a three ring binder with two disks tucked away in a plastic pouch at the back. The disks are not copy protected, but are inscribed graphically and electronically with the users name and serial number for reference. One disk contains the program while the other is named "data base" and is used for storage.

The opening selection menu contains what appears to be two documents represented by icons familiar to notepad users. These are actually dedicated start-up sequences meant to tell Caligari whether the user is operating with an Amiga 1000 or a 500/2000. Double-clicking on either writes the correct start-up sequence to your program disk. Then one double-clicks the left mouse button on the unassuming Caligari icon. After being presented with a high-res title screen, the screen blanks for a few seconds. The first workscreen that appears is called the Object Design screen. Instead of looking at a window of a world, you are looking through a window into a world. Before you stretches an XY grid, and in a short time you are made to feel much more like a sculptor then a graphics manipulator. You get the unmistakable feeling that you can reach in and touch the objects. The closest experience I can relate this to would be to assume the operation of the robotic hands that manipulate various substances behind protective glass panels. Quite a metaphor for microcomputer graphics!

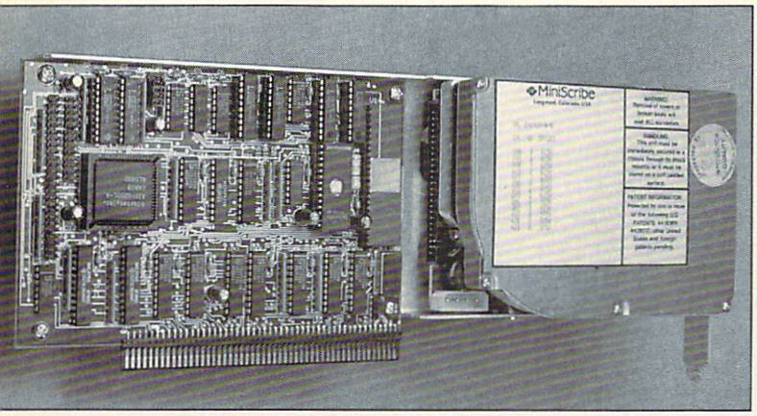
The screen is in high-res overscan (736 x 482). You could also opt to toggle the overscan off giving you the normal 640 x 400 screen. You could also operate in video-res (either 362 x 482 or 320 x 400). The manual mistakingly calls this "lo-res". At this moment in its operation, the program offers little in the way of anti-aliasing, so high-res seems the appropriate choice. These resolutions effect the rendering of the solid view displays alone, and are useful to consider if you want to save the image in IFF format for later manipulation in a paint or animation program.

One of the most useful features when it comes to utilizing a Calgiri sculpture in a paint program is centered upon the way the color render options are designed. You are free to have your object rendered in a very small amount of colors in a palette, but this does not mean the finished piece will have only those colors! No. It will invariably have a much wider range as a finished product. When I first saw the color menus, I had to reassure myself I was in Amiga's high-res mode because there were some

(continued)

HardFrame/2000

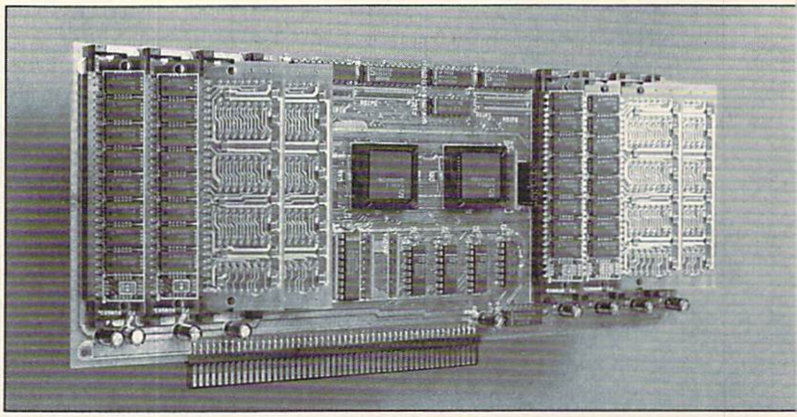
The Super-speed, DMA, SCSI Hard Disk Interface with 1.3 Autobooting



How fast is fast? **HardFrame/2000** transfers data at Amiga bus speeds! It's actually faster than the hard disk mechanism itself! And even more important in the Amiga's multitasking environment, **HardFrame/2000** has extremely efficient DMA circuitry to get on and off the bus in almost no time at all: 280ns to get on; 200ns to get off. **HardFrame/2000** autoboots under AmigaDOS™ 1.3 and is fully compatible with the new Fast File System. The core of any DMA SCSI interface is its SCSI protocol chip and DMA chip. MicroBotics has chosen the new, high performance Adaptec AIC-6250 SCSI chip, capable of up to 5 megabytes per second raw transfer speed, and the Signetics 68430 DMA chip running at 12.5 megahertz. Then we added additional FIFO buffering and enabled 16-bit wide data transfers for maximum throughput. The sophisticated design of **HardFrame/2000** provides for automatic SCSI arbitration, selection and reselection. The hardware supports either synchronous or asynchronous data transfer. **HardFrame/2000** can function as either the SCSI bus initiator or the target and can reside in a multiple master environment. Physically, **HardFrame/2000** is optimally flexible: the compact, half-size card comes attached to a full length, plated aluminum frame. The frame has mounting holes positioned to accept standard, 3.5" SCSI hard disk units such as those manufactured by MiniScribe, Seagate, Rodime, and others (hard disk mechanisms must be supplied by the user or his dealer as a separate purchase item). Alternatively, you can cable-connect to a SCSI drive mounted in your Amiga's disk bay or in an external chassis. As many as seven hard disks may be connected to a single HardFrame. There is no size limit on each disk. **HardFrame/2000** includes a 50-pin SCSI cable and header connectors for either 50-pin or 25-pin cable connection. Also included is a current tap to power frame-mounted drives directly from the slot itself. **HardFrame/2000** comes complete with driver, installation, and diagnostic software. Available September 1988. Suggested list price, $329 (hard disk not included).

The HardFrame/2000 photo shows the product with a MiniScribe 20 megabyte hard disk installed. Hard disks are not included in the purchase price of HardFrame. Note that if placed in the first slot, HardFrame uses only one slot.

8-UP! The Eight Megabyte Memory Card with Amiga™-specific DRAM Controller Logic



All the memory space you and your Amiga 2000 need –in a modern, highly integrated FastRAM expansion board. In **8-UP!**, MicroBotics went all the way to provide you with a truly Amiga-specific memory design to meet the special demands of the Amiga's high speed multitasking environment: The heart of any memory expansion is its *DRAM controller circuitry*. Rather than compromising with off-the-shelf parts, MicroBotics developed its own, custom controller design and built it into high-speed, *Programmable Macro Logic* chips (Signetics PLHS501). These new, super chips (each **8-UP!** uses two PML's) permit MicroBotics to employ *sparse refresh* technology to assure that your **8-UP!** is a truly zero wait-state/minimal-refresh-collision memory design. If you're putting eight megabytes in only one slot, that means that you probably have plans for your other A2000 slots. **8-UP!** gives you new freedom to do that planning since, unlike other ram peripherals, it is an extremely low-power memory card– a single, fully-loaded, 8-megabyte **8-UP!** draws an astoundingly efficient 800 milliamps! That's less than *two-fifths* of the power "budget" for a single slot! Low power draw also means that the card is cool-running for reliability and long life (not to mention a cooler Amiga!). **8-UP!** offers you maximum flexibility in memory configuration: it is organized into two separate *PIC's* (Amiga-speak for autoconfiguring peripherals). Each **8-UP!** PIC consists of four SIMM module sockets; these sockets accept either 256k-byte or 1 megabyte *SIMM's* (Single Inline Memory Modules). You can also purchase optional *PopSIMM* boards from MicroBotics; fill them with conventional RAM ; then use PopSIMM's to fill your **8-UP!** The card can run with as little as 512k of memory or as much as eight megs –with many intermediate configurations possible (particularly the six megabyte configuration, most desirable for use with a BridgeCard™). **8-UP!** is speedy, efficient, custom memory technology for your Amiga 2000 –and it's available now! **8-UP!** suggested list price is $199 (0k installed). Optional **PopSIMM's** are $49.95 per pair.

The 8-UP! photo shows the card half populated with conventional SIMM modules and half with MicroBotics PopSIMM's. PopSIMM's (without DRAM installed) are available as separate purchase items.

MicroBotics, Inc.

Great Products Since the Amiga Was Born!

811 Alpha Drive, Suite 335, Richardson, Texas 75081 (214) 437-5330

SOLD ONLY THROUGH YOUR AMIGA DEALER

Tell your dealer he can quick-order from MicroBotics directly – no minimum quantity –show him this ad!

"Amiga" is a registered trademark of Commodore-Amiga. "HardFrame/2000", "8-UP!", "PopSimm", are trade names of MicroBotics twenty-four separate colors on screen instead of the standard sixteen! What this means to the IFF user is astounding, although the awake and aware user has probably already guessed at how this is accomplished. I found it out when I dumped a creation saved in IFF format to DPaint for further rendering. Intermediate colors are dithered, utilizing mixtures of two adjacent colors in the palette. This means a full color model can appear on screen utilizing only two bit planes (four colors). This leaves you with the other twelve colors for more extensive painting. If you desire to utilize the dithered colors, just make a brush of an area of that color and use it as a patterned fill in a bounded area. The dithering is so exact that you have to use an on screen magnifier to appreciate it.

Of course, color rendering is the last step in Calgiri's design process. The first is the actual modeling of the sculpture under consideration. I've found that the first necessary step is to tell the program where your storage is to be directed and the path that must be taken to get there. This is the most difficult, in my opinion, operation in Caligari. I've already mentioned this to Roman, and perhaps others have too. Don't be surprised if your copy allows you a much more friendly pathway. Five separate kinds of graphic files can be saved and loaded at this time:

- *Objects-sculpted elements or parts libraries*
- *Polygons-shapes to be extruded and/or spun on an axis*
- *Scenes-compilations of glued together objects*
- *Animations-scripted wireframe or solids*
- *Lights-positions and options for light placement*

Caligari uses scripted animation files that can also be saved and loaded. They are created on a word processor of your choice, and a complete breakdown of the appropriate terms and syntax is notated in the manual. As of this first pre-release, the system only dumps the frame-by-frame animations to a single frame controller hooked up to a VCR, but an immediate release will address saving the frames in a compressed IFF format as well. Mr. Ormandy was very excited (as I am as an owner/user) concerning the release of Pageflipper+FX from Mindware International (about $150.00 US). This package allows magical manipulations in the IFF animation environment. For those of us unable at this time to afford single frame controllers and professional VCR's, this is a way to save Caligari creations to video.

Mr. Ormandy is adamant concerning wanting as few hierarchical menu selections as possible in Caligari, and has succeeded in that wish even in this first version. Non-hierarchicality means there is little branching in the user controls, everything being present on the screen in a block. This is accomplished by separating various menu choices into blocks so some operations can be accomplished with as little screen clutter as possible, giving you maximized views of your designs. He says "other software has hierarchically expanding menus the more options that are added. In the future, we will eliminate most of these."

At this writing, there are no print functions in Caligari although that will probably be remedied by the end of 1988 or sooner. I have successfully printed Caligari screens from the program by utilizing the program from IrseeSoft in Germany, TURBOPRINT (about $50.00 US), but was not able to print the 3D Object Design or Screen Design environments. Of course, you can print your finished sculptures from many programs once they are saved in the IFF format. Future upgrades of the software will also address plotter output. This only makes sense, as Caligari could prove to be one of the most professional micro-computer tools on the market for industrial design. Cost-wise and performance-wise, it will undoubtably find its way into the hands of engineers, mechanical and architectural designers, as well as the art and animation community. When the print function is added, it will also hopefully allow the user to dump the wireframe creations in the 3D environmental workspaces to the connected printing device listed in Preferences.

Two present methods exist for entering visual creations into Caligari's 3D world. One is a "primitives" menu which contains all of the standard 3D figures expected...cube, sphere, triangles, truncated figures...and there's room for more in the menu. The other method is the use of the "extruder". This choice presents the user with a grid (which may also be reconfigured). On the grid, the mouse is used to create a closed polygon in a step by step fashion. This polygon can then be extruded (given depth), used to make a cone like figure, or spun around either an internal or external axis in a lathe like operation. Caligari is so fast, I mean SO FAST, that you barely have time to breathe before your directions are translated into 3D artifacts in the workspace. Whatever this program benchmarks at must be astounding!

Once on screen, various elements in the design can be stretched or squashed in any or all of the XYZ dimensions. Then they can be glued together and unglued, and moved on, above, below, or very far away from the 3D grid. You can zoom in close enough to almost smell them and out far enough to almost lose them. In addition, you can spin and move the whole grid, including tilting it 360 degrees. The manual is clearly written, though I myself would prefer a smaller size because of limited workspace. The inclusion of an index would also be useful in a future version, as would be some representative graphics in the manual. Multi-tasking is not yet possible, but may be redressed in the near future.

Animation

As stated earlier, an instructional videotape is included with your purchase. On it are several animations that give you a basic idea of what Caligari can accomplish now, and where it's headed. One demo animation is also included on the disk, and you can preview all animated creations in wireframe mode. The biggest advantage Caligari offers as both a sculpting and graphics tool is speed. If you've got the right amount of memory (I'd say three megs or better, although you can function in a limited fashion with two) you're in good shape. Script files can be created on any word processor or text editor, and a very finite amount of terms produces a very complex animated sequence. A complete listing of the syntax is at the back of the manual. I am sure more "animation scripting primitives" will be offered in the future, i.e., pre-designed scripts that will take your creation and manipulate it in a pre-edited fashion. Once you design and proof your script, it can be used with any sculpted creation if it is desired.

(continued)

There are a million great reasons to write an Amiga Program.

This year, The Disc Company will spend more than one million dollars acquiring new Amiga software programs to market around the world.

Which is great news for all the programmers and product designers hoping to cash in on the software they've developed.

The programs we're looking for need to be innovative. And they must meet the highest quality standards. We're especially interested in games, audio and music products, video programs and graphics software, but any original product will be considered.

Once they've been accepted, programs will be translated into eleven or more languages, and marketed worldwide through our international distribution network.

As the world's fastest growing Amiga software publisher, we'll make sure your product reaches more people, in more countries than you ever thought possible.

If you've got a program you'd like us to consider, call or write us for a free Software Submission Kit. All the details will be enclosed.

Oh, and thanks a million.

In the United States and Canada:
The Disc Company
3135 South State Street, Ann Arbor, Michigan 48108
Phone 1-800-456-8418 or (313) 665-5540
Fax 1-313-761-9586

In Europe:
The Disc Company - Europe
1 Rue du Dôme, F-75116 Paris, France
Phone (33) 1-45.53.10.53
Fax (33) 1-47.04.62.35

In Australia, Asia & New Zealand
The Disc Company - WordLine
P.O. Box 404, Round Corner, NSW 2158, Australia
Phone (61) (02) 484.8086

THE DISC COMPANY

Amiga is a registered trademark of Commodore International, Ltd.

Presently, I am using Caligari in two very different ventures. In the first, I am creating an illustration for a national diskazine (JumpDisk) for a scene from Dicken's Christmas Carol. In this scene, Old dead Bob Marley, chains and all, is approaching the open door to Scrooge's bedroom. With Caligari, I have designed the hallway, walls, bannisters, and downward stairway in Scrooge's house. I will then render this in 3D after getting the proper angle of view, and paint other elements into the scene. The second project I'm involved in concerns a design idea that a friend of mine has for a new commercial aircraft. I'm going to render it in 3D, spin it around in an animation sequence, and videotape it for the proper agency in Washington D.C. Calgiri will make this possible without any of the upgrades that can only improve its current magic.

The biggest battles Mr. Ormandy faces in convincing users in the bulging-at-the-seams Amiga 3D/Animation marketplace to invest thousands of dollars in Caligari centers, it seems to me, are on three issues: Comparative cost, IFF compatability, and the finished smoothing characteristics of one's work. At this time, you are paying for the use of Caligari's unbelievable interface and speed. As long as no-one else releases a similar design, Caligari will indeed attract users who lust after its speed and ease of use. But given the history of competitive fervor in the industry and the rock-star accolades great software designers receive, who can say how long Caligari will remain alone on the peak of Mt. Interface. As to IFF options, Caligari is at a serious disadvantage at this time. All of the other vendors offering professional 3D/Animation tools at least allow the user to input IFF backgrounds, and four that I use also allow for IFF brush-wraps on 3D objects. Caligari has to redress this as soon as possible for those of us to whom the Amiga is more than a toy. As for the current smoothness of the finished product, Caligari isn't. Mr. Ormandy promises a quick fix in this area, and the quicker the better. Polygonal shading is apropos in a very limited number of situations.

The Future

Roman Ormandy can tell you about near and far plans to upgrade Caligari, some as close as weeks away and others two years down the road. I'll just list a representative number of them to give you the flavor of what's ahead in Amiga graphics engines:

- IFF input as well as output
- VLAN & IFO animate
- Full color rendering (16 million colors with the proper hardware)
- Addressing a Targa board on an Amiga 2000
- Free upgrades for a year to purchasers of the pre-release version
- A complete graphics/animation workstation with the Amiga as the central artifice for under $10,000
- New primitives in the extruder environment
- Marketed libraries from other users
- Dedicated materials and surface textural rendering
- Suggestions as to professional hardware interfaces
- State-of-the-Art desktop videography

The list is even longer then this, but you get the idea. The name of the game is support and continuance, as well as designing and marketing an original offering. The computer graphics multiverse will be a lot different five years down the road, and I'm sure Caligari will be playing a very central role.

No matter what your decision regarding a purchase of Caligari at this time (although if you are a professional designer in any area that utilizes computer graphics and the Amiga, I would urge you to jump on board) I would plead with you to purchase the demo disk from Octree (about $18.00). It does everything that the pre-release version does except to save your creations, and if nothing else, you'll have a lot of fun. Here's a few words gathered in an interview with Roman Ormandy himself...

"Caligari has no competition. We will present a family of products in the future. Human 3D communications is our business. Imagine Caligari moving solids rather then wireframes, simulation rather then animation...interactive microworlds and new forms of communication. Octree will provide social simulations in interactive 3D. We will be working upon human problems with 3D simulation products, work stations hooked up to fiber optics networks, FTDN networks within five years. Business applications will be introduced more and more to 3D applications. The Macintosh opened up a whole new way of dealing with information in a 2D way, and now the Amiga is a bridge to a more advanced computer architecture. Amiga is the best current video-output computer for 3D graphics."

But Commodore, take heed. By his own admission, Roman Ormandy is not an Amiga fanatic. He is just using the best system currently available, and will port his wares to other dedicated systems in the future. If Commodore doesn't stay on top of the heap, it will be surpassed, and developers like Roman Ormandy will market their wares elsewhere. Is Caligari for everyone? Obviously not. Is Mr. Ormandy's philosophical stance an indication of a brave-new-world mentality? Perhaps. Only time will tell.

Editors Note:
An updated version of Caligari will be out by mid-December. It contains specific troublesome data base fixes, and a completely revised look and feel for the file requesters. It also boasts being enabled to access the 68020/68881 configuration which significantly increases modeling and rendering speeds. A number of commands have been added to the animation module as well. Both VLAN and VSIO are now fully supported.

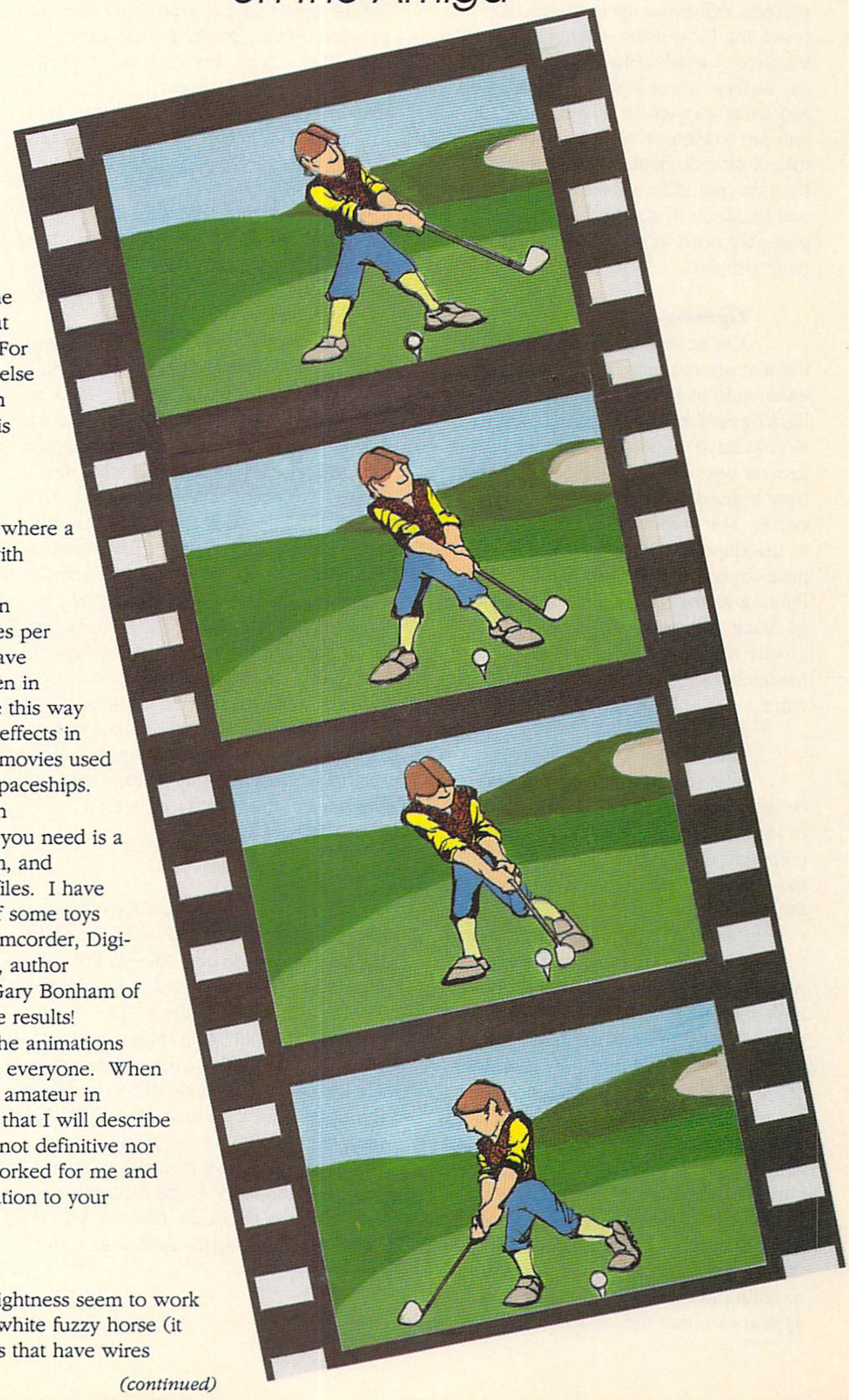
Caligari
OCTREE SOFTWARE
311 W.43rd St., Suite 904
N.Y., N.Y. 10036
Pre-release version: $2000

Create Your Own...

Stop-Motion Animation
on the Amiga

by Brian Zupke

Most of you are aware of the Amiga's graphics, sound, and specifically, animation capabilities. There are many animation packages available for the Amiga, from Aegis Animator to Antic Zoetrope. Some of these packages generate objects from a set of coordinates while others use IFF pictures. The problem I have with the latter packages is that I'm not an artist and cannot draw very well. For those of you who are like me (and everyone else too!), there is an animation technique that can bring about amazing results. This technique is called stop-motion photography.



Stop-motion?

Stop-motion photography is a process where a static scene is photographed over and over with minor changes made to the scene between photographs. When the pictures are shown in sequence at a fast rate (typically 8 to 24 frames per second), the objects in the scene appear to have motion. Stop action photography is used often in animation. Many short films have been made this way using clay models. It is also used for special effects in motion pictures. For example, the Star Wars movies used stop-motion photography to animate all the spaceships.

With very little equipment, stop-motion photography can be done on the Amiga. All you need is a video camera, video digitizer, a paint program, and programs to create and play back animation files. I have created several black and white animations of some toys and other household objects using a video camcorder, Digi-View, Deluxe Paint, MakeANIM (PD program, author unknown) and ShowANIM (PD program by Gary Bonham of SPARTA, Inc.). I was really surprised with the results! Enough people have asked me how I made the animations that I decided to share what I've learned with everyone. When reading this, keep in mind that I'm merely an amateur in animation and photography. The techniques that I will describe are for black and white animation. They are not definitive nor will they work for everyone, but they have worked for me and hopefully you will be able to use this information to your advantage.

Choosing a subject

Objects that have a high contrast in brightness seem to work fairly well. I had no luck whatsoever with a white fuzzy horse (it showed up as just a white blob). Rubber toys that have wires

(continued)

inside, such as Gumby and Pokey, are excellent for body movement. I used a "Noid" rubber toy in my animations with fantastic results.

Choosing a background

Using a black background and surface works best. This way only your subjects will show up in the picture, providing for quicker editing and smaller IFF files. I used a black cloth to cover the surface where I placed my subjects and used another for a back drop. If you are working with a dark subject, then it may be better to have a white background although the IFF pictures will be larger in size and you will probably need to do more editing in a paint program.

Lighting

Use at least two lights and place them at approximately 45 degrees on either side of the camera (this is the best lighting method for digitizing pictures). Do not have a light source close to the camera because your subjects will not have a good contrast causing a loss of detail. The amount of light (brightness) to use depends on your subjects. What I have found to work best is to set the lighting so the background shows up as all black (the background color). This greatly reduced the amount of "retouching" I needed to do with Deluxe Paint.

Camera

The video camera you use should be placed on a tri-pod or on some sort of stand. It is important to have complete control over the camera and its movement. Having the camera point at the exact same location requires less editing later. I used a Sony 8AFU 8mm camcorder for the animations and it worked fairly well. Although it doesn't have the resolution the B&W security cameras have, it is convenient to use since it has a viewfinder. I can quickly see what is in my field of vision.

Having a zoom lens is also handy. I can make my subjects appear to move towards or away from the camera by changing the amount of zoom over a series of frames. Special effects in motion pictures are often done by having the subject remain stationary while moving the camera to give the appearance that the subject is moving.

Script

Before you start to digitize a sequence of frames, it is a good idea to write up a script or storyboard describing what kind of action is to take place. If you describe or sketch each action in terms of frames, then digitizing the action is simplified. For example, I wanted my "Noid" toy to spin around in a circle, and decided that 16 frames were enough to show that action. For each frame of the spin, I rotated the subject 1/16 of a circle. Without a script, it's easy to get lost and forget which direction objects are supposed to be moving.

Digitizing Procedure

Once you've got a script and have set up some sort of stage (where all the scenes will be shot), make adjustments to the focus and lighting. These adjustments should be made before you begin saving digitized pictures to disk. Digitize test pictures of some of the key scenes in your script to fine tune the set.

Now that you have everything set up, you're ready to start digitizing the scene. For each frame you digitize, write down any important information about it, such as necessary editing. Save each frame with the name of the scene and a frame number (example: noid1.bw, noid2.bw, etc.). This makes it easier to keep track of the frames you've digitized. While each frame is being saved to disk, start making the changes to the scene for the next frame. This will reduce the amount of time it takes to digitize all the pictures. Continue this process until all the frames you want have been generated.

Editing

Once all the frames have been saved, you may want to create an ANIM file before you do any editing, just to see how the sequence turned out and what kind of editing is required. The "MakeANIM" utility is the easiest way to create an ANIM file. First, create a text file listing the names of all the frames in the order you want them shown (include the file pathname of the pictures if necessary) and save it to disk. If you want the animation to be a continuous loop, be sure to list the first two frames in the sequence again as the last two frames.

Run the "MakeANIM" program by typing:

```
MakeANIM sequence name.ANIM
```

where "sequence" is the name of the file containing the frame sequence you've just created, and "name.ANIM" is the name you want the ANIM file to be saved under. The "MakeANIM" program will load in each picture and display it on the screen. It then calculates the differences between the current picture and the previous one and saves this information in the ANIM file. When the ANIM file is complete, "MakeANIM" will report how many frames were processed.

Now use "ShowANIM" to see your animation. Type:

```
ShowANIM -l +5 name.ANIM
```

for non-continuous loop play or type:

```
ShowANIM -c +5 name.ANIM
```

for continuous loop play. The "+5" specifies that each frame will be shown for 5 jiffies (1 jiffy is 1/60 of a second).

Version 5.3 of ShowANIM allows you to control the animation while it is playing with the following keys:

F1 to F6 -
set frame delay to Fn jiffies per frame

F7 -
set frame delay to 10 jiffies per frame

F8 -
set frame delay to 20 jiffies per frame

F9 -
set frame delay to 30 jiffies per frame

F10 -
set frame delay to 60 jiffies (one second) per frame

down arrow - pause animation
up arrow - continue animation
right arrow - show next frame
left arrow - restart animation
C - switch to continuous loop mode
L - switch to non-continuous loop mode

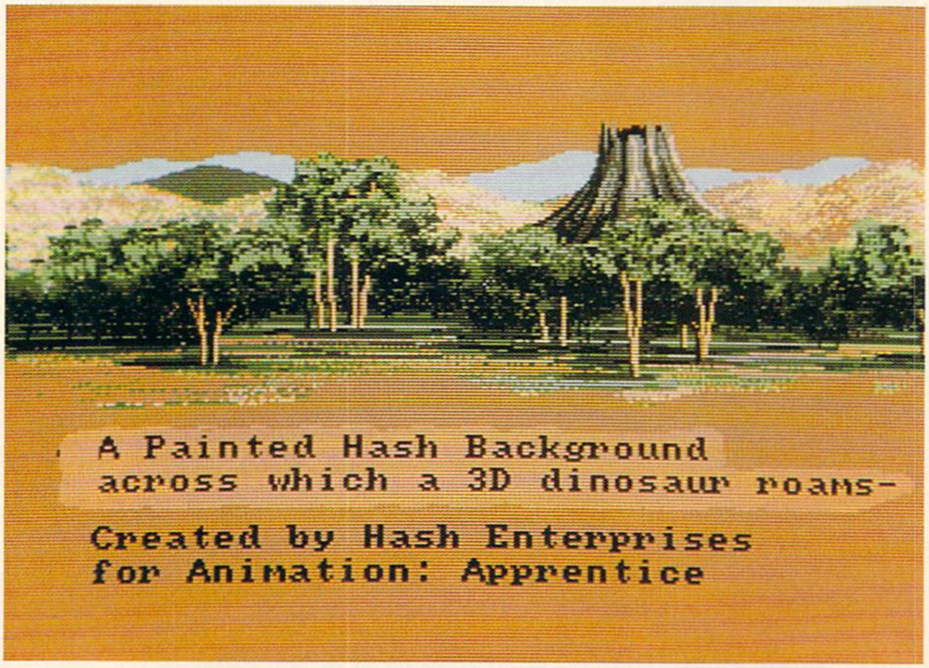
(continued on page 31)

The Wonderful World of Hashnique

by R. Shamms Mortier

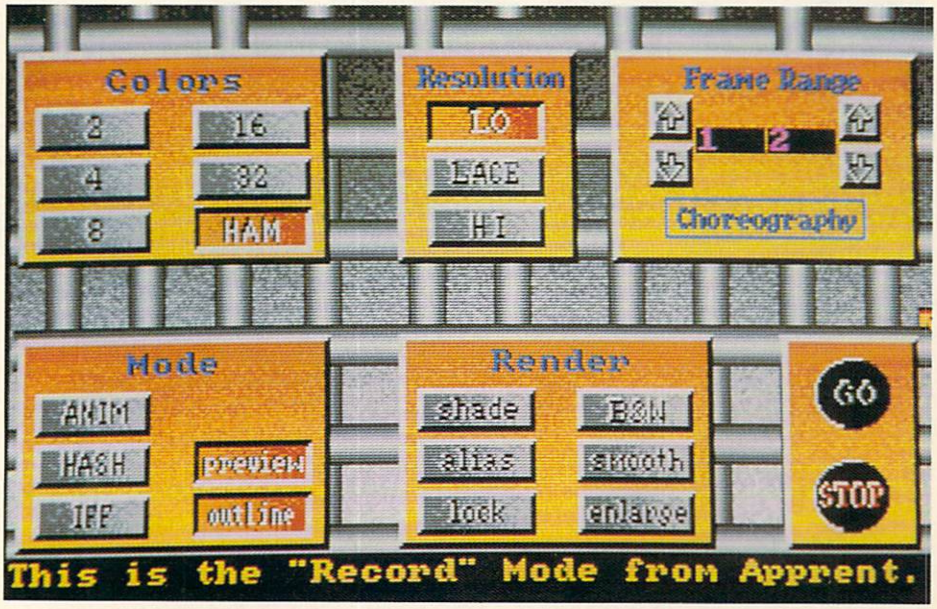
A review of the Amiga animation software products of ***Hash Enterprises***

If you are considering doing animation on your Amiga, any kind of animation, then you would do well to investigate the numerous software packages offered by Hash Enterprises. If you skip over these on your way to animating your fantasies, then you truly do so at the expense of your own creative potential and capacity for mastering microcomputer animation. A few years from now, if their creative potential and determination hold out, Hash Enterprises will stand in that rarefied company at the top of the animation developers mountain. Even now, the ripples of their efforts are beginning to make waves that wash over all Amiga animators, and their work is pushing the "expected" and the "possible" to new states-of-the-art. Don't take all of this laudatory verbiage as gospel. Judge for yourself. Investigate the products and you'll "see" what I'm saying.



A Painted Hash Background
across which a 3D dinosaur roams-

Created by Hash Enterprises
for Animation: Apprentice

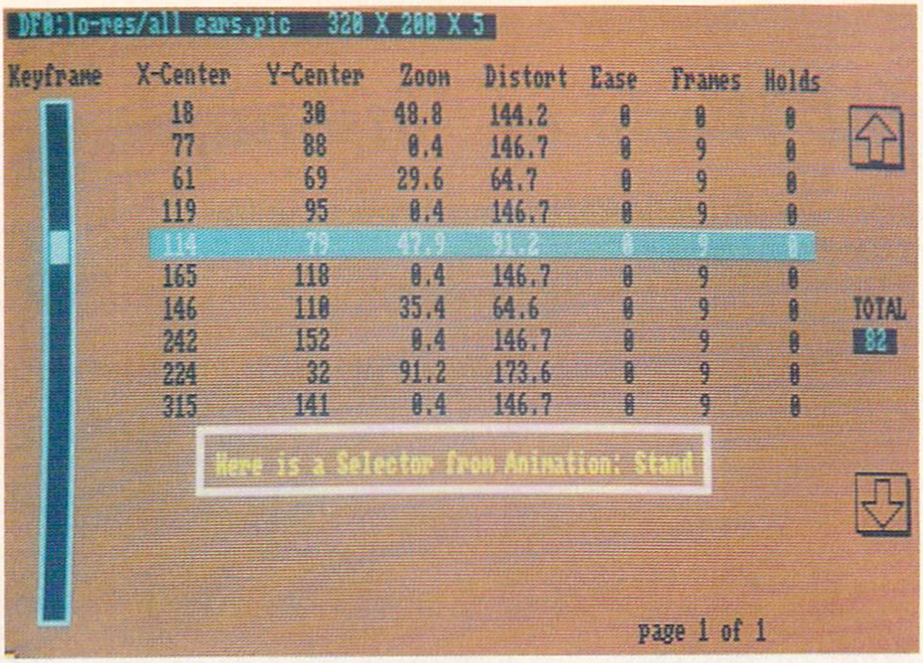


This is the "Record" Mode from Apprent.

Hash-History

To hear Martin tell it, he's been an addict of the Saturday Morning animated madness since he was eight or so, and even expanded upon that obsession in his teen-age years by amassing a huge comic book collection. His grandfather was the recipient of many of Martin's early graphic works. He admits to "living for things animated", and at one time was a guest on the Art Linkletter show because of his interests and habits. Later on he received his B.A. in Electrical Engineering and an M.B.A. in business, to be enhenced still later by a Masters Degree in Computer Science. After a stint in industry as a "nameless cog in a giant machine" where he observed that his soul was being lost, he started to find his way back to those life adventures that had always attracted him: Animation and the arts.

The return actually began back in 1981, when with "visions of floating spheres" he contacted a special effects



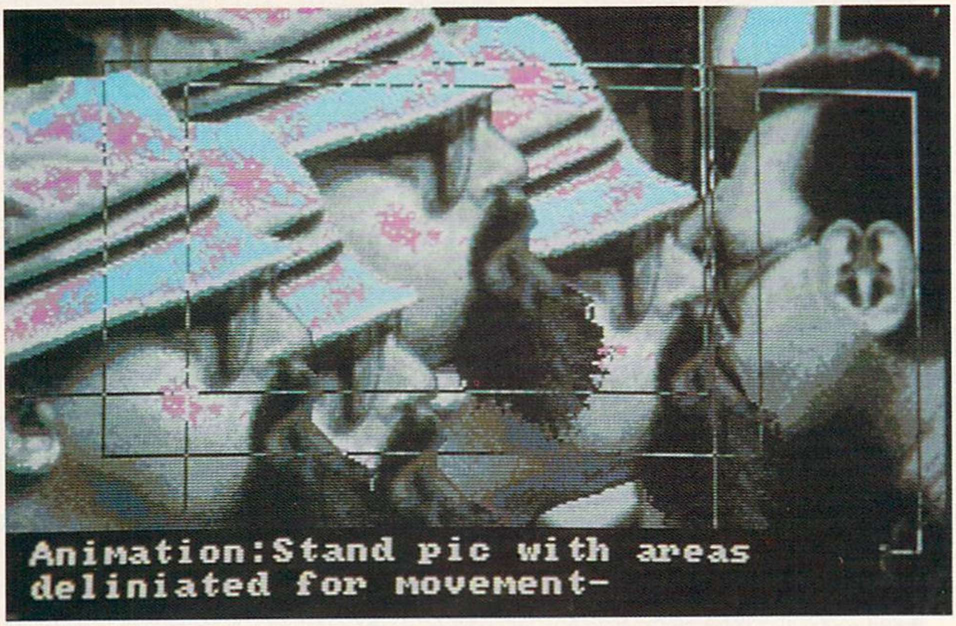
DF0:lo-res/all ears.pic 320 X 200 X 5

Keyframe	X-Center	Y-Center	Zoom	Distort	Ease	Frames	Holds
	18	30	48.8	144.2	0	0	0
	77	88	0.4	146.7	0	9	0
	61	69	29.6	64.7	0	9	0
	119	95	0.4	146.7	0	9	0
	114	79	47.9	91.2	0	9	0
	165	118	0.4	146.7	0	9	0
	146	110	35.4	64.6	0	9	0
	242	152	0.4	146.7	0	9	0
	224	32	91.2	173.6	0	9	0
	315	141	0.4	146.7	0	9	0

TOTAL 82

Here is a Selector from Animation: Stand

page 1 of 1



Animation:Stand pic with areas deliniated for movement-

producer in Portland, Oregon. Through this association, he met Gene Ellis (animator, photographer, cameraman) and they founded a business that lasted some four years. They created a special camera which produced and filmed animation and special effects. "Computer animation was just beginning and I jumped right in". It was at this time that he envisioned attributes of a system that would eventually find their way into Hash's Animator: Apprentice software..."a character animator like no other...treating each character as an actor in a play, each of whom had dialogue, a script, and direction...I formulated the concept of a Scene Editor".

Although he believes "mechanical animation" has its place, his first concern is to preserve the fifty years of animation history that have allowed the audience to identify with the animated characters on the screen. The heart of "Apprentice" is the modeling section, based on a process he unashamedly calls "Hashnique". It allows you to construct 3D fully painted organically perceived models, creatures that run, walk, crawl, and behave in legitimate "alive" ways.

Contact!

My introduction to Martin's work came from my response to an ad placed on the back page of one of the haughtier Amiga magazines. It was a rather small unobtrusive announcement considering its content, obviously designed without the help of some slick agency. What it offered was simple..."Disney style animation on the Amiga". No beating around the bush or qualifying statements. Just that. At that time, the program was one of the most expensive on the marketed as far as graphics and animation. $500.00. It was called "Animator: Apprentice" and my lust was ignited. There was a phone number listed (and good thing, because without one, I would not have considered the purchase. Phone numbers assure me, rightly or wrongly, that the vendor has some degree of stability and potential for longevity). Martin himself answered the call. In a world where "service" often means being shunted from one level of ignorance to the next, I was actually speaking with the developer himself. He gave me all of the details that I needed to take the plunge. Service with a capital "S".

That was a little more than a year ago, and at that time the "line" of Hash products was a monafilament line,..Apprentice and nothing more. As you will discover from my description of the products that they now offer, (and read "now" as "available now" and not as "vaporware") though their product line has become a veritable rope, intertwined products that answer needs that many computer animators didn't even know they had. Products that handshake with each other and also with other non-Hash works. When I listen to Martin speak about his work, I get the impression that I'm listening to an excited teenager ready and able to revise history and the way things are done. I've never heard the jaded overworked crotchety beleaguered entrepeneur in his voice, so I'll have to assume that those all too common traits are not a part of his persona. I've always suspected that he leans to the artist side of the artist-scientist personality.

Animation: Apprentice

This first offering from Hash Enterprises still stands out as the most unique, and remains at the center of the entire product line. The present version as of this writing is 2.0, and it has been revised and upgraded too many times for me to list, each time at a minimal cost to owners. This is the most unique animation engine on the Amiga market. Period.

Not that you can't design those "floating spheres" with this package, but that's not really its main purpose. Martin wasn't kidding when he offered the Amiga community the chance to design, render, and animate Disneyesque characters. Within the limits of the resolutions of the machine, this is quite possible. The heart of Apprentice gives the Amiganimator the unheard of capacity for not only creating, but storing these creature parts in a library for future use. They can then be used to produce features limited only by the creative extent of the animator's mind and her/his access to the right recording equipment. How close is that to your Amiganimation dreams?

The whole operation is tied together in a fairly complex hierarchical system, one that requires a good amount of study and experimentation. My main complaint is that the manual doesn't offer as complex a tutorial as is necessary in this quest. Perhaps this is because Animation: Journeyman, the next phase of the task, is soon to be released (Spring 1989?), and all creative juices are flowing towards that end. I don't know. The Apprentice manual also has no index, and that's an oversight needing redress. Those of you familiar with other reviews/articles that I pen, recognize this to be an ongoing complaint I have with many of the Amiga packages, especially the more complex ones. All manuals MUST have complete indexes! Users should demand this as their birthright! One last negative and then on to the mashed potatoes. The present manual is a reduced version of the first one, which came in an oversized binder. Obviously, the original was put on a copier and reduced to its present size. This made its production more economical for Hash Enterprises, but for folks like me with the ocular capacity of a snail, it makes it much more difficult to read. Hopefully, this will not be the case with Animation: Journeyman.

The very soul of Apprentice's engine consists of the Object Design Module. Here, one designs unique body parts. For my previous complaints concerning the manual, I should also state that it progresses very clearly through each of Apprentice's stages: Sculpt (Object Design creation/editing of 3D "parts"), Character (relating or building the full objects), Action (defining the parameters and boundaries of motion that an object may display), Director (defining the speed of an objects movement across a stage), Rehersal (running through a test of the final animation), Record (the finished rendering of the piece), and Display (playback). It is in the Sculpt section that Apprentice best demonstrates the ingenuity and creativity of Martin Hash. Objects are called "segments", discrete body parts. You load in a "visage" (front silhouette) and a "profile" (side silhouette), which you have previously manufactured in a dedicated paint program. On top of this, a texture map (surface features) is placed. The computer then combines the totality of this information into a 3D object, which, when married to other objects, represents a dedicated character. The non-textured form is called a segment blank, and is composed of 3D slices computed from the silhouettes, much like present scanners in hospitals are able to generate visualizations of internal organs and processes through non-invasive techniques. Specific palette locations are demanded for foreground, background, and shaping colors, and must be adhered to in the paint program design stage. Varying the axis of spin will allow shapes to bulge and protrude at

(continued)

SOFTWARE SENSATIONS

inky dink
ink-jet ink for printers
specially formulated for the Xerox 4020 and Diablo c-150 printers.

3 sizes, 4 colors.
tested over a year

half the price of the manufactures ink.

Brummbar's Backgrounds

a series of IFF painted backgrounds, brushes and objects. Brummbar is recognized throughout the industry as a master of the Amiga Palette.

S-view a cable interface connecting Y-C video to S-video (super vhs, ed beta) VCR's and monitors

S-VIEW

1441 SO. ROBERTSON BLVD.
LOS ANGELES, CA.90035
(213) 277-8272

THE HOTTEST AMIGA STORE IN THE COUNTRY.

Amazing Computer Systems is *hot.* Our sizzling selection of Amiga products has become the talk of the town. From more than 1000 software titles in stock to the hottest selections of hardware, accessories and books. All at red-hot prices. So remember. When you're hot, you're hot. And when you're not, you're not shopping Amazing Computer Systems.

Amazing Computer Systems, Inc.
Frankford Village Shopping Center
3030 N. Josey Lane #144 (2 doors South of Skaggs) Carrollton, Texas 75007 (214) 394-8383
Open Mon.-Sat. 10am-6pm Thurs. 10am-8pm
Amazing Computer Systems is not affiliated in any way with PiM Publications, Inc. or Amazing Computing.

the sculpt stage. A special touch-up procedure is offered after texture mapping occurs, and the object is ready to be saved for future participation in the final character design.

Character implementation is next. It is in this phase that much familiarity with the program is required. Characters can be composed of one or of several "figures", and several levels of object hierarchy are possible. Attributes such as "siblings" and "progeny" ad infinitum can cause you to lose your way if care and attention are lost. Each segment can represent one "bone" of a completed figure, and if care isn't taken, strange unexpected Frankensteins can result (has Martin checked with the FDA and AMA for the appropriate license to practice digital genetics?). Everything can be open to Editing, Culling, Divorce, Adoption, and Placement, giving you the tools for very complex character development.

In "Action", you are allowed to define the movement parameters of a character. This is a great screen, and you might spend many hours here without ever considering moving on. Here is where you witness the amazingly lifelike movements of a stick figure moving in real time from front, side, and top views. You can tweak the angles of selected bones, adjust the starting-ending positions of the figure, and have it engage in bouncing and stretching actions. You are advised to keep a chart of the actions, as they can quickly become extremely complex. It would, in fact, be smart to keep dedicated journals of all of the procedures you experiment with for future reference and post-edits.

"The Director" allows you to control the choreography of a scene. This is done through a combination of camera, stage, and viewplane movements. All results can be previewed. "Rehearsal" allows you a preview of the entire choreography, including playback at Frames-Per-Second (FPS) ranging from 4 to 60. The maximum rate might not be achievable if your work is too complex. Then "Record" does what it says. You can either "pack" the frames for maximum storage, or render them in IFF for further treatment. You can choose the color range for 2, 4, 8, 16, 32, or HAM. Shading can be added so that the object responds to light sources and specular attributes, and overscan is also possible. "Display" is the final choice, allowing you to speedup or slowdown the action during playback, or to view it one frame at a time. It can also be played backwards.

This package can be immensely valuable to the Amiga animator, and is truly a precident setter. The present cost is $299.95.

Animation: Flipper

"Flipper" animation is becoming a generic term in Amigaland. Many vendors produce Flipper style programs in various levels of difficulty and application. The easiest type (of which the Hash version is an example) is good for "pencil testing", pre-checking a series of frames before render and record operations. The most optional for comparison would probably be PageFlipper+FX from Mindware International (retail: $150.00). Considering the price of Hash's Flipper of $59.95 and its options, it is a good package for the price. Resolutions supported are of the IFF type, not ILBM, so that HAM is not available. All other resolutions and parameters are supported including overscan, although you should take care not to mix resolutions or palettes in any one sequence. Two types of "packing" are offered: ANIM files and special HASH files (for use with other Hash programs). The technique is simple. Load and display a number of IFF frames or paintings, speed of playback being somewhat determined by the complexity of the images. A range of 4 to 30 FPS is possible. The manual is adequate and clearly written. This is the easiest program in the Hash series.

Animation: Effects

I have made good use of this package in the production of some very involved titling and instructional videos at the University of Vermont. At a suggested retail price of $49.95, this software is a good bargain. Once you learn the interface operation, which is a cousin to Apprentice, you can create very complex choreographies useful for fancy titling applications. Although not as seemingly sophisticated as VideoEFX-3D from Innovision Technology, which operates exclusively in HiRes (retail for about $150.00), "Effects" will allow you to utilize any of the Amigas IFF modes including overscans. Wipes, scrolls, and

You thought Microfiche Filer was just for *WIMPS?**

Now it's for *PROGRAMMERS*, too!

The easiest-to-use database has just become the most powerful with the introduction of **Microfiche Filer Plus™.**

Microfiche Filer Plus™ includes a full programmer interface using ARexx. Its 33 powerful new commands allow database access from any ARexx macro.

In addition to Microfiche Filer's lightning sort and search facilities, Microfiche Filer Plus™ adds:

- ☐ Automatic Field Calculations
- ☐ HAM Picture Storage
- ☐ Database Programming
- ☐ Direct Control of Other Applications
- ☐ Dynamic Data Import and Export with Other ARexx Compatible Applications

*Window, Icon, and Mouse Persons!

SOFTWARE VISIONS Inc.
P.O. Box 3319, Framingham, MA 01701

Use Microfiche Filer Plus™ to:

- ☐ Control your terminal program and store your downloads
- ☐ Index your disks and execute programs directly
- ☐ Cut and paste data to such programs as TxEd or TeX
- ☐ Store and execute ARexx macros in the database

Only a REAL PROGRAMMER can appreciate the full power of Microfiche Filer Plus™!

MICROFICHE Filer™ PLUS

Only $179. (ARexx sold separately)

For your nearest dealer call 1-800-527-7014.
In MA call (508) 875-1238.

flips can be jammed into a choreographic listing and applied to any chosen screen or brush, depending upon the quantity of RAM your machine has. The larger the matte though, the slower the display is likely to be. I like to work with smaller text brushes and genlock them for the final result. Like VideoEFX-3D, the finished work is saved to a special data disk one frame at a time for later replay. Unlike VideoEFX-3D, this package also offers you the option of adding a light source and variable shading possibilities. Packing options accompany the saving process, and IFF backgrounds can be included. Acceleration of frames from one to another can be individually controlled. The 68020/68881 is advised and supported. "Effects" is really three separate programs: Wipe (allows you to generate various screen transitions), Scrolls (great for scrolling credits and other graphics), and Flip, the spinning rolling functions. For the money, this program is a bargain. The manual is a bit harder to read than "Flipper", but clear tutorials are included.

Animation: Stand

There is an opening scene is Disney's Pinnochio that moves the view from a night sky, through the streets of a small village, right to the puppet maker's doorway. This whole scene was painted on one giant canvas, and then the camera was moved accordingly. All of this work was meticulously accomplished on an animation stand, and Hash's "Animation Stand" can be utilized in much the same way. One experiment that I hope to attempt in the future is to use ExpressPaint (P.A.R.:$100.00) in this process. ExpressPaint allows the user to define a giant screen size that can be scrolled in steps, so that a very large painting is possible. Imagine floating over an ever changing background. Hash has made it possible. The manual suffers from too small a type size, but it's clear enough to get you addictively involved.

Animation: Multiplane

This is one of Martin's latest offerings, and answers many animators needs in the Hash environment. It retails for $89.95. The old definition of a "multiplane" relates to the Disney multiplane camera used for the first time in the production of "Bambi". It consists of several movable layers or plattens upon which artwork is placed. The camera shoots through each sandwiched layer that has transparent areas so that the underlying layer can be seen, allowing for amazing depth and action. Hash's Multiplane allows the Amiga animator much the same alternatives. Depending upon your machine's memory constraints, you can shoot through your frames stacked background colors from one level to the next. One Meg is required, and 2.5 is needed for HiRes stuff.

The manual is the best of the Hash series so far, clearly written and well documented in large bold type, with adequate tutorials sprinkled throughout. Various effects can be produced that push multiplane into new experimental areas: Color Control alters the RGB color guns in any animated sequence; Contrast sets the threshold values; Brightness allows you to transit from a black all the way to a white screen; Dissolve fades one layer into another; Image Offsets

(continued)

Hypertext

for AMIGA (512K)

Unleash your creativity
Organize your thoughts

THINKER

An Outline Processor &
A Word Processor &
Picture Display with
Hypertext links to
Text and Applications

THINKER helps you write programs, books, and articles or organize reference material, pictures, and documentation.

Introductory Price $59
No Credit Cards, CA Res. add tax, 30 day guarantee
Until 4/89

Poor Person Software
3721 Starr King Circle
Palo Alto, CA 94306
415-493-7234

controls the XY presets; Pixelate changes the width/height of pixel boundaries; Polarization remaps the pixel colors; Strobe/Motion Blur simulates speed effects; Shadow allows you to drop shadows on the back layer; Wipe offers sophisticated transitions. All changes can be controlled on a frame by frame basis as well as globally. I have only begun to test drive this package, but already question how I was able to do without it.

And On And On...

The Hash Enterprises Animation Newsletters are packed with upgrade information, hints and tips for beginning and advanced animators, and special pricing on new and revised products. Also supported are the animators that design and submit addendum products. All of Hash's future works will be copy protected using the "key disk" method, and who could blame him, considering the state of rip-offs in the industry. I wouln't like to have authored this stuff only to have it appear on some teeny-bopper's bulletin board for downloading. Extensive testing and listening to user feedback assures the user that all of the Hash line will be constantly updated and revised at a very nominal user charge, usually just $5.00 per disk. There is even an automatic update program whereby users are sent the upgrades automatically and billed for the difference. Miss one payment for a long enough interval, and no more service for you. Just tell them you want to sign up and send in serial numbers of present Hasheries. The current newsletter (volume two number four) also includes a list of commands allowing you to run everything from the CLI. I'm convinced that Hash and Co. want to offer their version of every possible program that can be used in the CompuAnimation environment. Let me hint at some of the releases that will either be available when you read this, or that are just around the corner...

Animation: Editor ($59.95)
Edits files created for display. Included are conversion routines that allow for handshaking with other animation formats and products.

Animation: Rotoscope ($79.95)
Hand modify previously generated animations. Load a background animation into the computer and draw into an overlay animation. Supports all resolutions including HAM, and includes a special HAM black and white conversion module. Allows HAM backgrounds for normal 32 color overlays.

Animation: Soundtrack
(Beta release on Dec.1, 1988)
Synchronizes soundtracks with animations. Promised to be simple and intuitive to use, yet offering powerful results.

Animation: Quick 2D
(no release date set yet)

Animation: Journeyman
(Spring 1989? Upgrade from Apprentice around $100.00)
The much awaited professionally oriented upgrade to Apprentice.

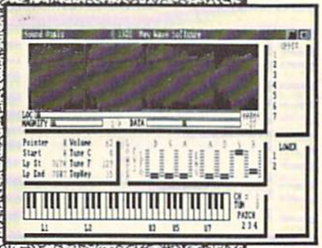
The Farther Future

Martin Hash dreams about being a major player in the transition of animation from film to video, and from the expensive studios to the home workspace. To this end he has met with Bill Hanna himself, and has gathered some disquieting facts that will effect this transition. For one, he learned that most of the work done on the Saturday Morning cartoon shows is done in Asiatic markets, and that the animators are paid about 25 cents a cel! That's about fifteen dollars for a full days work. Can you imagine an American professional animator working for that? Not hardly! This means ways to produce more cels in a day than currently possible must be found, and that's where the computer comes in. That means better hardware and new standard peripherals for all interested parties. Of course there's a built in conundrum here. Instead of bringing work like this back home to the U.S., the same Asiatic markets may opt for the same technology. Well, I'm sure that whatever happens, Martin will be there somewhere, marketing his state-of-the-art wares.

Other dreams that spin in his animated consciousness, concern utilizing 3D animation in the 3D stereoscopic realm and also in holographic work. He is already engaged in supplying animated sequences for a new generation of video games, and in working up new 3D scenes for the Viewmaster stereoscopic viewers. Not to be caught with all eggs in one basket, Hash Enterprises is porting its wares to IBM, Atari, and Apple systems.

They also offer an expanding series of character disks at $24.95 each. Now available are Dinosaurs, Dragons, Knights, and Santa and Friends. Two disks of Geometric shapes are also offered. Anything purchased direct from them gets you a 20% discount. Many contests are available for entry, and a professional film group is supported and open for your enrollment. He continues to exhort users to pay attention to the aesthetic and artistic elements of their work, instead of relying soley upon the technological miracles. This is a refreshing change than the opinion that the technology is an endall in itself. If you've been bitten by the Amiga animation bug, scratch the itch by investing in the Hash line of animation helpware.

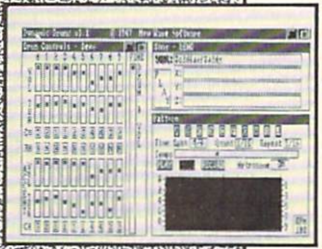
Hash Enterprises
2800 East Evergreen
Vancouver, Washington 98661
(206) 693-7443



Sound Oasis

$99.95

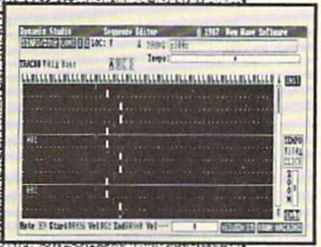
The internal sound capabilities of the Amiga are better than that of any other personal computer. These capabilities mean nothing though, without quality digital sounds, which up till now have been scarce. Sound Oasis gives Amiga owners access to a large library of studio-tested digital samples, by using the Amiga's built in disk drive to read disks made for the Mirage Digital Sampling Keyboard. Sounds can then be played from a MIDI keyboard, the computer keyboard, or saved as an IFF standard file. Mirage is a trademark of Ensoniq Inc.



dynamic Drums

$79.95

Transform your Amiga into a professional-quality drum machine with this software package. Easier to use than hardware-based drum machines because everything is displayed graphically on screen. Enter drum patterns quickly and easily in real time with visual feedback and editing. Create realistic drum tracks with any of the 100 drum and percussion samples that are included or use your own unique IFF one-shot samples. Dynamic Drums also has full MIDI implementation and even becomes velocity sensitive when triggered from a MIDI keyboard.



DYNAMIC STUDIO

$199.95

A powerful MIDI sequencer that takes full advantage of the Amiga's sound, graphics, and sophisticated user-interface. Dynamic Studio is perfect for professional applications due to its sophisticated editing capabilities and SMPTE support. It is also ideal for home studios, because in addition to sequencing MIDI instruments, Dynamic Studio has a built-in drum machine, and the ability to playback instruments translated with Sound Oasis.

No Copy Protection

NEW WAVE SOFTWARE

P.O. Box 438 St. Clair Shores, MI 48080 (313) 771-4465

(***Stop-Motion***, *continued from page 24*)

Make notes of editing required while playing the ANIM file. Some of the frames will probably have unwanted background pixels that will need to be removed. If there are many of them, you may need to remove these before you create your first ANIM file simply because the file will be too large to fit in memory. This depends on how much memory your Amiga has.

It's possible that a subject's location in a frame or a series of frames will need to be moved to make it a smoother animation. This is easily done in Deluxe Paint II by making the subject into a brush, erasing it, and moving it to its desired position. A nice thing about DP II, is that you can load two pictures into memory at once and flip between them by pressing the 'j' key. This allows you to check two consecutive frames for continuity. If you do this, you may want to type the frame number in one of the corners on each picture — it's very easy to forget which is which. Once you're ready to save the pictures to disk, remove the number from the picture.

If your scene requires an object to be held up on a stand or hung by wires, these items can be easily removed from the frames with a paint program.

Non-digitized graphics, such as props, scenery, or special effects can be added to an animation by using a paint program also. If you add graphics to the animation and are using "MakeANIM", be sure to use the same color palette for each picture of the animation. "MakeANIM" cannot handle a change in the color palette during the animation.

If you have a genlock device, you can show your animation on top of a video background. The possibilities here are endless!

For digitizing color animations, I imagine a lot more fine tuning of the set would be required. The lighting and color of the background become much more important when working with color. I haven't experimented with color since my camera produces terrible digitized color pictures.

The ANIM file format is only one of many methods of creating animation on the Amiga. There are dozens of animation and graphics programs (public domain and commercial) that can be used to help create animations with stop-motion photography. So much more can be done than just black and white animations. You can add color, digitize in color, add graphics from other programs, use a genlock device, and add sound. Use your imagination! I know that I've only scratched the surface of the very tip of the iceberg! Let's see what our Amigas can do! Happy digitizing!

Deluxe Paint II
Electronic Arts
1820 Gateway Drive
San Mateo, CA 94404
(415) 571-7171
$99.95

Digi-View
NewTek, Inc.
115 W. Crane St.
Topeka, KS 66603
(913) 354-1146
$199.95

•AC•

Second Generation

by Geoffrey Williams

(Top and Bottom)
Scenes from an animation created with Photon Cel Animator.

From the moment of its unveiling, the Amiga was billed as a powerful animation machine. The first generation of 2D animation software hinted at great promise, but was far too limited in speed, resolution, and color. The second generation of animation software has arrived, and it fulfils much of that earlier promise.

While the first generation of software seemed geared to the home user, the second generation is aimed solidly at professional animators and the educational and corporate video markets. These new products have undergone extensive Beta testing by video professionals who have used the products in projects for their clients. With the feedback from these serious users, developers have been able to create some very stable and powerful programs.

There are two basic types of 2D animation software for the Amiga: Cel Animators and Key Frame Animators. Cel Animators load in and manipulate individual images created in other programs, flipping through them so quickly that the illusion of motion is created. Key Frame Animators do the animation, you just have to provide the key points of movement.

In this article, we will take a look at the new cel and key frame animation programs, how they differ, and how they are being used by professional animators. Most of these products were reviewed in pre-release versions, so there may be some minor changes and additional features in the final releases. They should all be on the shelves by the time you read this.

Cel Animators

In traditional cel animation, the animator draws black and white line art of his backgrounds, characters, and other objects using a special punched registration board that lets him keep all of the images lined up with each other. These are either drawn on or xeroxed onto clear acetate (originally celluloid, hence the term "cel"). He assembles

2D Animation Software

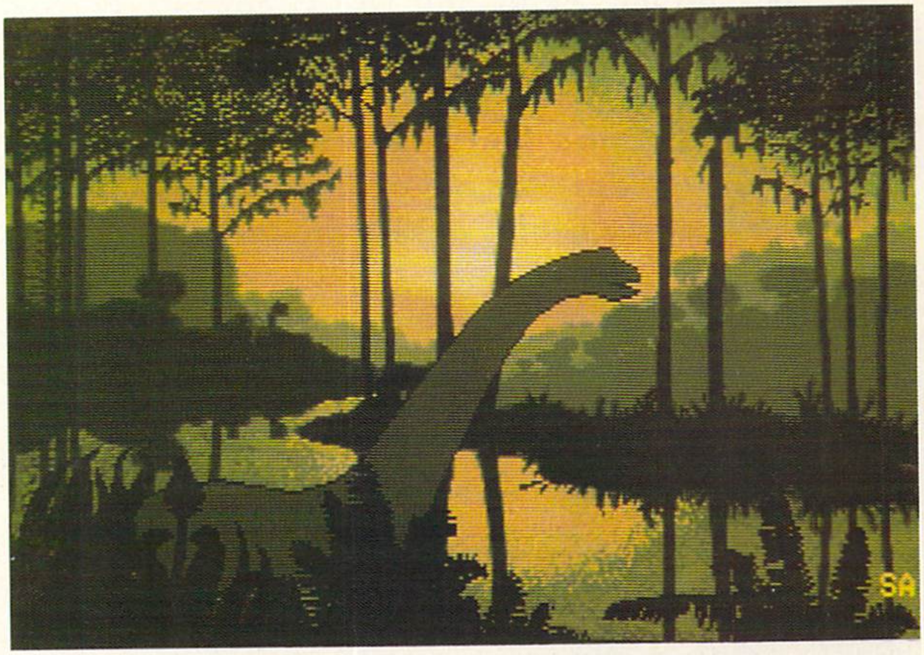
Animating with Photon Cel Animator, Pageflipper Plus FX, Animator:Flipper, Backlight, Zoetrope, and Fantavison.

these elements in layers for each and every frame, and shoots them with a video camera a frame at a time. The animator can then play the video to watch the animation, noting any changes that need to be made. After making changes, he must shoot the entire animation all over again, since video must be shot from beginning to end. This is called pencil testing. When all necessary changes have been made, the animator colors in the cels, reassembles them, and shoots the animation all over again. Besides the arduous process of assembling each cel and capturing it on video, the animator must also worry about dust, scratches and fingerprints getting on the cels.

With a computerized Cel Animator, the artist can use any Amiga paint program to draw each cel. When all of the cels are loaded into the Cel Animator, he can play them in any order, at different speeds, add new cels, and play with color palettes. The advantage is instant feedback, as the changes are made,the animation can be viewed immediately. When any change is made in an animation using traditional techniques, the process of video taping each frame must be done all over again.

While individual features may vary, Cel Animators all have in common one basic function. They load many individual images into memory and flip through them very rapidly, just like the cartoon flipbooks you played with as a kid. The standard frame rate for video is 30 frames a second, which means you may have as many as 30 individual drawings for each second of animation. Even for fairly simple animations, you should have at least 2 megabytes of memory.

A way to save memory is to use delta compression techniques that save only the changes in pixels between each frame. This is the technique that is used in the Anim format, a compression standard officially recognized by CBM and supported by many animation programs. While generally used to save



Scenes from ***Fantavision*** *(top) and* ***Zoetrope*** *(bottom) animations.*

(continued)

Erich ***Stein & Associates, Inc.***

Public Relations Consultants

Because the quality of your reputation is just as important as the quality of your product.

PO Box 695
Denver, Colorado 80201
TEL (303) 733-3707

animations in single compressed files, some programs allow you to manipulate compressed animations in memory. This saves a lot of memory, but there is a trade-off in speed and capability.

Photon Cel Animator

Part of the Photon Video line of products published by MircoIllusions. Programmer Jim Donaldson has added many features that make this a useful tool for animators. It will handle images in any resolution including HAM and overscan (although all frames must be in the same resolution and color palette, a general limitation of this type of program). It has a special pencil test mode that will turn full color IFF images into two bit planes, making the images black and white and allowing you to get a lot more frames into memory. Pencil test playback also ghosts previous images to let you see paths of motion.

One of PCA's strongest features is the ability to aid the animator in synchronizing sound. This has been a difficult task for animators, either requiring a lot of trial and error, or you have to slow down the tape so that you can distinguish how much sound is in each frame. It is very difficult to figure out what you are listening to when the tape is slowed down this much. Traditionally, the sound track is created before the animation, and you follow the same procedure with PCA. Load the digitized IFF sound samples into memory, and synch them up to specific frames (which at this point will still be blank). You will then be able to play each frame and hear what sound is associated with that frame in real time. You can enter a three letter code, such as a phoneme, into each frame, and get a printout of an exposure sheet which includes these three letter codes. You then use the exposure sheet as a guide when drawing the animation so that each frame is drawn exactly for the sound associated with it.

For animations larger than memory, Photon Cel Animator is designed to work interactively with Photon Video Transport Controller (also by MicroIllusion) so that you can record each image to video tape frame by frame. Have both programs running, set your frame rate from within Photon Cel Animator, and let the Transport Controller automatically dump each frame to tape. You also need a frame by frame controller such as the Lyon Lamb Mini Vas, which is the actual hardware that the Transport Controller controls.

Photon Cel Animator supports the Anim format. By saving an animation as an Anim, you create a single file of your animation which can be run independently by using the public domain PlayAnim program. The compression can be quite dramatic, depending upon how similar the frames in your animation are to each other. I borrowed a 15 frame HAM animation that took up three disks (almost two megabytes). By saving it as an Anim from PCA, the resulting file was only 300k. PCA will also load in Anims, even those created by other programs, and decompress them so that you can make changes.

There are some simple drawing tools, which would be handy for doing touch-ups, but are certainly no replacement for a good paint program. Instead of drawing on the computer, some animators prefer to sketch out each frame in pencil, then digitize them. The area fill and palette controls would be useful for coloring digitized line art.

There have already been several commercial projects created with PCA. Ernie Potvin has used it to create animations for point of sale video displays. Adventures Unlimited, under the direction of Tom Miller, created a full length animation to promote a water purifying system. Two short animations from this are included on the two animation disks that come with the package. One is of a flowing river with waterfall, and the other of a fairy splashing water on her face from the stream.

The program is easy to use, and very straightforward. In most cases, you have a choice between pull down menus or keyboard equivalents. Its biggest downside is that it is very memory hungry. The more megabytes you have, the better. Although the manual was written by a professional animator Heidi Turnipseed and is very clear and informative, the program is easy enough to use that you will be able to jump right in.

Pageflipper Plus FX

In contrast, you will have to spend a lot of time with the PageFlipper Plus FX manual. As one user remarked, "It is a deep program". PFX takes an interactive script approach, reminiscent of the Director although not as versatile. You can write the script in a word processor, but you are far more likely to use the interface that allows you to click on various options to edit and create the script.

Even with the interface, the process takes a little learning. You select your frames from a file requester, which automatically creates a script for you. On the left side of the screen there is a scrolling window with a script and gadgets that you use to edit the script. To make changes, you must choose the type of change you want to make by selecting a gadget, click the area in the script you want to change, and then select various options from a pop-up menu. While this seems easy, there are many nested commands which you must use the proper syntax and the proper order. The script uses a display language, and you must have an understanding of the language to use it. A major saving grace is online help. Click on any gadget, menu option, or command in the script with the right mouse button, and a contextual help file will pop up.

The user interface is not difficult to master, once you understand the script structure. It is inelegant in appearance, but the manual points out that this was a compromise to save memory. Pop-up menus replace pull down menus for the same reason.

PFX works only in compiled mode, meaning that animations are always compressed in memory. This saves large amounts of memory, although you have to remember to recompile the animation after every change. Recompiles are very fast.

The advantage of a script language is flexibility and power. PFX lets you dynamically adjust individual colors, speed, and dimming through sixteen levels of color as an animation plays. It also has a single step mode, and an interactive mode that uses the function keys to trigger animations and effects. You can create loops, and load in brushes to save memory. You can flip entire animations upside down or backwards.

Individual frames can trigger Intermmezzo (sequential) or slave (parallel) animations. Slave animations run concurrently with the animation that triggers them. This works because both animations are running in different bit planes at the same time. Since this is a major feature of the program, the manual has an in-depth explanation of how bit planes work, and this knowledge is necessary to utilize this feature.

Slave animations are very useful for special effects, and this is where the FX of the program comes in. There are 76 single bit plane effects on the disk, including a variety of wipes. They will not work with HAM or half-bright images though, since they use all available bit-planes. Using these set effects is far from automatic. You must go through quite a series of menu options, making several changes to the script in the proper order and in the proper place.

The claim of Anim support is somewhat misleading. PFX will not save in Anim format. It saves in a proprietary format, which is supposed to be more efficient than Anim. I tested two simple animations, one saved in Anim from Photon Cel and one saved in PFX. The results: HAM Anim- 320k, HAM PFX- 449k, Hi-Res Anim- 105k, Hi-Res PFX- 385k. I suspect there is a lot of overhead in the PFX format, and that it might well be more efficient if you have a lot of looping or special effects. For more straightforward animations, it is much less efficient. PFX will not directly read in Anims either. Their sole Anim support is a separate public domain utility that decompresses Anims into IFF files which can be saved to disk and individually loaded into PFX.

PFX has been in the hands of beta testers for a year, and Canadian production company ani/FX has used it to create an animation to show the names of Uniroyal and B F Goodrich coming together for a series of employee training videos that dealt with the merger of the two companies. Pixelight created an animated commercial for C.I.L. to advertise a weed killer, and the commercial was aired during the Stanley Cup Playoffs. They combined digitized images with DPaint graphics, and the entire project, from storyboard to tape, was completed in 24 working hours.

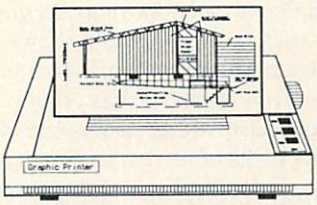
Animation:Flipper

For the cost-conscious animator, Hash Enterprises has been turning out a series of low cost animation tools. Their page flipping product may not have a lot of frills, but it gets the job done.

In one module, you select the frames you want to use from a file requester. Once the order and basic parameters are set, the program will load in the images, compress them, and save them as either an Anim or in Hash format.

FOR SALE

Hi-Resolution HP Pen Plotter $50 *



PlotPrint works in conjunction with all CAD and scientific programs that produce great-looking plots. It translates the HPGL pen-plot output to extremely high resolution printer output. **PlotPrint** works with any graphic printer: ink-jet dot matrix and laser. It supports color, scaling, and rotation and has a Preview feature. **PlotPrint** requires an Amiga, with 2 drives and minimum of 1 Meg RAM, and your graphic printer.

Order: #PP-U $50 Standard Amigas
#PP-T $100 68020/68881 Amigas
Check, money order, or COD only. Include $2.00 S+H. Dealer inquiries are welcome.
Order from:

Hi-Tech Graphics
P.O. Box 446
Tallmadge Ohio 44278

*Software Product---No Plotter Required.

Once saved, you play the animation through the player program, which works with the entire line of Hash animation products. The player has pause and function key speed control, and gives you an easy way to control playback of the animation.

If you have other Hash products, this is designed to integrate well with them. Its low cost makes it an excellent entry level product, and it may well do all that you will ever need.

Backlight

Backlight started life as an in-house development tool for Silent Software. It was originally conceived by Reichart Von Wolfshield, and evolved under his direction with primary programming by Leo Schwab (of cape and Workbench display hack fame). Reichart also gives special credit to animator Eric Daniels for helping with the fine tuning. Eric worked on Disney's "Who Framed Roger Rabbit?", and he created the walking chrome Sculpt 3D

(continued)

legs that made the rounds on the BBS's. Since there is nothing like Backlight on the market, and a great need for a tool of this type, Silent Software decided to make it available to Amiga artists and animators.

Backlight is designed to aid the artist in drawing an animation, as well as in animating it. One of its key features is the ability to layer drawings. As you draw each frame, Backlight shows you the sequence of previous frames in successively lighter shades of grey. This is enormously useful, as you can clearly see the paths of motion as you draw, and you can use the previous drawing as a guide to your current drawing.

You are not stuck with crippled drawing tools either. A lot of effort went in to making the program as DPaint compatible as possible. There are the familiar drawing tools (although in the early version of the program I saw, they were available from pull down menus), the keyboard shortcuts are the same, it uses the same brush concepts, and even has a J screen that is available throughout a sequence of drawings. You can cut and paste, and even append sequences of an animation.

While Backlight works in all resolutions, it only supports the two color mode. This is not really a limitation, since animations are generally created as line drawings, and colored later when the sequence is complete. When you have the animation the way you want it, save the files in IFF format, load them into DPaint, and add the finishing touches.

One of its most astounding features is real time compression and decompression. Backlight keeps all of the frames stored in compressed form in fast memory, decompressing them on the fly so they can be displayed. This happens so fast you do not even realize it, giving you all of the memory savings of compression, with none of the speed losses. You never have to wait for an animation to recompile. You can also save animations in this compressed form, or as individual IFF images.

Backlight also supports the film frame rate standard of 24 frames a second, a feature Photon Cel Animator lacks. In fact, you can set any frame rate, or choose from set speeds of 30, 24, 15, or 12 fps. When an animation is playing, you can instantly start or stop it.

Beta tester Rusty Mills has worked as an animator for just about every major shop, doing work on projects from "Sport Goofy" for Disney to "American Tail" for Don Bluth. He now works freelance, and is using Backlight to develop an animation series for Fox television called Latidga (an anagram for digital). Rusty bought his Amiga because of the potential of Backlight after being introduced to it by Eric Daniels.

Rusty first designs his backgrounds in Sculpt 3D, animates them in wire frame, and imports them into Backlight and draws his character movement over the backgrounds. The backgrounds are rendered in Animate 3D, and the characters and foreground objects are colored in DPaint. Color testing is done in Photon Paint, and final assembly will be done by Genlocking foregrounds over the sculpt background.

With its speed, ease of use, and thoughtful design, Backlight will become an important development tool for animators. It has almost everything you could want in a basic animation tool, including features I did not think possible. With its very low cost, no animator should be without it.

Key Frame Animation

In key frame animation, you draw the beginning and end point of a specific movement, and the computer fills in all of the images in between in a process called tweening. This is what most people think of when they think of computer assisted animation, but in many ways it is inferior to cel animation. The artist does not have the same level of control that the cel animator has, and the animations tend to be less rich as complex objects are much more difficult to control.

Having the computer do the animating for you can save an awful lot of time. With careful planning some very high quality animation can be achieved. Both Deluxe Video and Aegis Animator, the first two animation programs for the Amiga, did tweening. The following two key frame animators go far beyond them in capability.

Zoetrope

The zoetrope, invented in 1831, was one of the first animation players. By spinning a wheel and looking through the slots, the images inside appeared to move. Zoetrope is also the name of the animation package written by Jim Kent, the author of Aegis Animator. Zoetrope combines the flexibility of cel animation with the ease of use of key framing. Its one major drawback is that it provides only lo-res output, which gives it the lowest quality output of any of the programs in this article. This does limit it for many professional applications, but it has enough features to justify a serious look.

With a number of simple drawing tools, you can create the images in your animation frame by frame. Objects from the last frame turn blue and act as a guide as you draw the next frame. With the history command, you can easily do what animators call in-betweening. You draw the frames with the most significant action first, and then keep drawing between those frames to smooth out the animation. Using the history command, you can draw a new frame while seeing the previous and following frame in blue. You also can edit, delete, insert, and duplicate frames, as well as adjust the timing.

You do not have to animate frame by frame. Once you draw an object, or load it in as an IFF file, you can do key frame animation. Draw a box around the object, creating what the author calls a clip, set its position in the start and end frames, and choose the type of effect you want to create. These clips are just like Deluxe Paint brushes in that they are easily moved and manipulated and can have transparent areas. You have many options under several categories: Pixel effects, moves, and APM fx. You can make a clip rotate around its x, y, or z axis (although not in 3d) as it enlarges and moves across the screen. You can trace a path, even one with a lot of curves, and the clip will be animated along that path. You can choose two types of anti-aliasing, which really cleans up clips as they go through movements. You can distort clips in several ways, or do unusual special effects with options such as crystallize, which builds up a pattern of colored pixels over an object. Play this in reverse, and it would create a Star Trek like materialization of an object. Effects can also be generated that affect the entire screen, and you can do color manipulations and fades. It would take a lot of space to describe all of the effects, and they can be combined in dozens of ways.

Unlike Animator, which moved Polygons in real time, Zoetrope manipulates bit maps. When you setup

PHA$AR™

New! Version 3.0

The best personal accounting package for the Amiga has just gotten better.

Introducing PHA$AR version 3.0. Updated to take full advantage of the Amiga's friendly user interface, PHA$AR now incorporates more powerful analysis, reporting, and financial functions.

PHA$AR 3.0... the only way to make sense of your personal finances.

New Features:
• Enhanced online calculator with 5 level memory buffer
• Sorting by account name or number • Improved tax program now doubles as report writer • Custom define screen colors
• Recall most recent transactions • And many more.......

General Features
• 130 expense/income categories • 40 accounts (VISA, checking etc.)
• Up to 500 transaction per month • **Artificial Intelligence** parser predicts input, minimizes typing • Instant context-sensitive help function • Relational design integrates financial data

Transactions
• Create transaction macros for repetitive entries • Point-and-type spreadsheet-style relational summary table

Reports and Analysis
• Analyze loan/saving plan • Clearly formatted printed reports include: Account summaries—Category summaries—Net worth statement—Tax calculations • Sort categories in any order • Built in data graphing

Tax
• Completely integrated tax preparer and report generator.

Date/Address Organizer
• Online organizer stores up to 250 names, addresses and phone numbers • Integrated scheduler stores special occasions. • Event alarm feature

PHA$AR 3.0
$89.95

Antic Software

Available from:
Antic Software
544 Second St.
San Francisco, CA 94107
(800) 234-7001

an animation effect, it creates each frame that is a part of that effect. This means that each frame of an animation can be saved as an IFF file, whether the computer animated it or you drew it frame by frame. Not only that, but you can use the Over and Under commands to merge animations together. You can create a tweened animation of a moving logo, and create a separate cel animation of a walking man, then combine the two separate animations into one. You can even position the starting point of objects with the mouse, and have the man walk behind or in front of the animated logo. This is a powerful and flexible feature. You can also set loop points and run animations backwards, or back and forth in ping pong mode.

The one major impression I have of this program is that it is fast. When you fill a shape with a color, it happens almost instantaneously. An entire screen fill takes about half a second. It also generates the frames of an animation with remarkable speed. Changes are fast and easy, and there is tremendous flexibility built into it.

Despite its limited resolution, Zoetrope is being used in professional applications. Animator Steve Segal, of "Dance of the Stumblers" fame, created an animated scoreboard for a Canadian commercial which was played in enlarged mode to create the proper pixilated look. For the film "Face of the Enemy", Steve created a simulated computerized library catalog system with different sections of text that scrolled underneath moving highlighters.

One of the features Steve really appreciates is the ability to load in Animator files. Zoetrope automatically converts them into bit-maps. The moving searchlights in the Zoetrope picture started out as objects in Animator. The background and sky was created in Deluxe Paint, while the theatre is a composite of two digitized images. Zoetrope can also load in the .rif files created by the Live digitizer.

One of the standout uses for Zoetrope is as a fast animation sketchpad, allowing you to quickly experiment with a wide variety of effects and looks. Even if the final product needs to be at a higher resolution, Zoetrope would be a great tool to use when you sit down with a client and rough out ideas. Since it uses both computer animation techniques, it would also be an excellent tool to work with to teach yourself basic animation techniques.

Zoetrope is not Anim compatible, but you can use the provided Player program, or dump your animations to videotape. I suggest that you use one of the public domain programs such as Lacer or DoLace to change the screen to interlace without changing the resolution. This gives you a better quality image on tape.

I suspect that a program with this much power, but with higher screen resolution, would outrun the chip memory capacity of the current Amigas. It would also make it one of the best tools on the market. As it is, the resolution is limiting, but I think it will find many useful applications in the professional community.

(continued)

Fantavision

Brodurbund's Fantavision is a pure interpolation program, and does not generate individual frames between key frames as Zoetrope does. For doing quick and simple animations in almost any resolution, including HAM and Overscan, it is a very useful tool.

Fantavision allows you to draw squares, circles and freehand objects with multiple points. The points can be moved and dragged to change the shape of the objects. It is very similar to the polygon objects of Aegis Animator. You can also easily add or delete points, do rotations, size changes, and numerous other manipulations to these objects. Each object is numbered, and you can clone the key frame and its objects to make the next key frame, or you can make a blank key frame to create new objects. Similarly numbered objects will metamorphise from their shapes in the previous frame into the new shapes. You can also cut and paste objects from frame to frame. By pressing the GO gadget, you can play your animation at any time.

A simple example would be to draw a circle in key frame 1 on the right side of the screen, and a square in key frame 2 on the left side of the screen. You can set the number of tweens between key frames (up to 128) and set the animation speed. Play the animation, and the circle moves across the screen as it changes into a square. You are not limited to a single object either. In fact, you can have up to 16 objects per key frame.

You can also load in IFF and HAM backgrounds, which can be changed without effecting the animation. To create an animated bar chart, I loaded a grid created in DPaint to use as a guide, drew small boxes at the bottom, and in successive key frames I moved each box up to the appropriate position while using the Trace mode. Trace moves the object while tracing an image of it behind. I then changed the background to an IFF image of a company logo. In five minutes I was able to create a very effective animation of growing bars representing company sales.

IFF images that you have loaded in can have sections of them cut out, like brushes in DPaint, and these cutouts can then be animated to move around the screen (although their basic shape cannot change). They can be mixed along with the metamorphic shapes, and even layered so that objects can pass behind or in front of other objects.

Fantavision supports Colorfonts (although fonts are not metamorphic objects), and it is easy to use whatever fonts you have available. A requester allows you to set the directory for fonts, so you can keep them on a separate disk.

Even though the metamorphic shapes you create are essentially single colored, Brodurbund provides 38 preset palettes to work with (or edit your own). Most of these palettes have textures and patterns to improve the appearance of your objects. Objects can also be combined together to make more complex objects.

One of the most impressive features of the program is its ability to incorporate digitized sounds and even "tween" them. There are nineteen digitized sounds on the disk, or you can use your own IFF sounds. A sound requester allows you to adjust the volume, balance, echo, pitch, duration, and looping of each sampled sound. I drew a large image of a rocket in key frame 1 and had that trigger the thunder sound at full volume. In key frame 2, I reduced the size of the rocket and moved it to the top of the screen, and used the same sound but set the volume to low. When played, the rocket takes off and the roar of the engines diminishes as the rocket gets further away. By setting the balance controls and changing them from right to left between key frames , you could have the sound appear follow an object across the screen in stereo. You can have up to four sounds going at the same time.

Fantavision is fast and easy to use. It seems to me that it particularly shines as an educational tool, allowing you to quickly and easily create animations to illustrate key concepts. You can have an IFF image of an engine as a background, with animated moving parts over it. You could create moving molecular structures, or show the flow of electrons in a circuit diagram. There is a simple but effective demo on the disk showing the side view of the head of a Cro-Magnon Man, which progessively transforms itself into the head of Modern Man. It uses only two key frames, and clearly illustrates the power of metamorphic objects.

In the dinosaur picture, the background was created in DPaint, while the dinosaur is a metamorphic object that lumbers across the screen and stops to nibble some metamorphic leaves. The ferns in front of the dinosaur are clips created from a DPaint image, and the dinosaur moves behind them.

Although it does not approach the quality of true cel animation, Fantavision is well thought out and very intuitive to use, with many options and features. It is not Anim compatible, but does include a distributable player program. You can create some very effective animations with it, and you'd have to use it quite a while before you explored all of its capabilities.

There are two public domain programs you should also add to your animation tool kit. MakeAnim takes a list of frames and turns them into an anim file, and the latest version (5.0) offers tight compression and sound. GrabAnim works as a background task with any program. Hook up a joystick or mouse to the joystick port, and every time you click the button, whatever is on the screen will be added to the Anim file that the program generates. This turns DPaint into a much more efficient animation tool, as the additional step of saving each frame and then compiling them is eliminated.

No single program described in this article is the ultimate animation tool, but by using them together you should be able to do almost anything you need to. The Amiga is powerful enough to create professional quality animation, and with the second generation of animation software, you have the tools available.

•AC•

Product Information

Photon Cel Animator
$149.95
MicroIllusions
17408 Chatsworth St.
Granada Hills, CA 91344
(818) 360-3715,
Outside CA (800) 522-2041

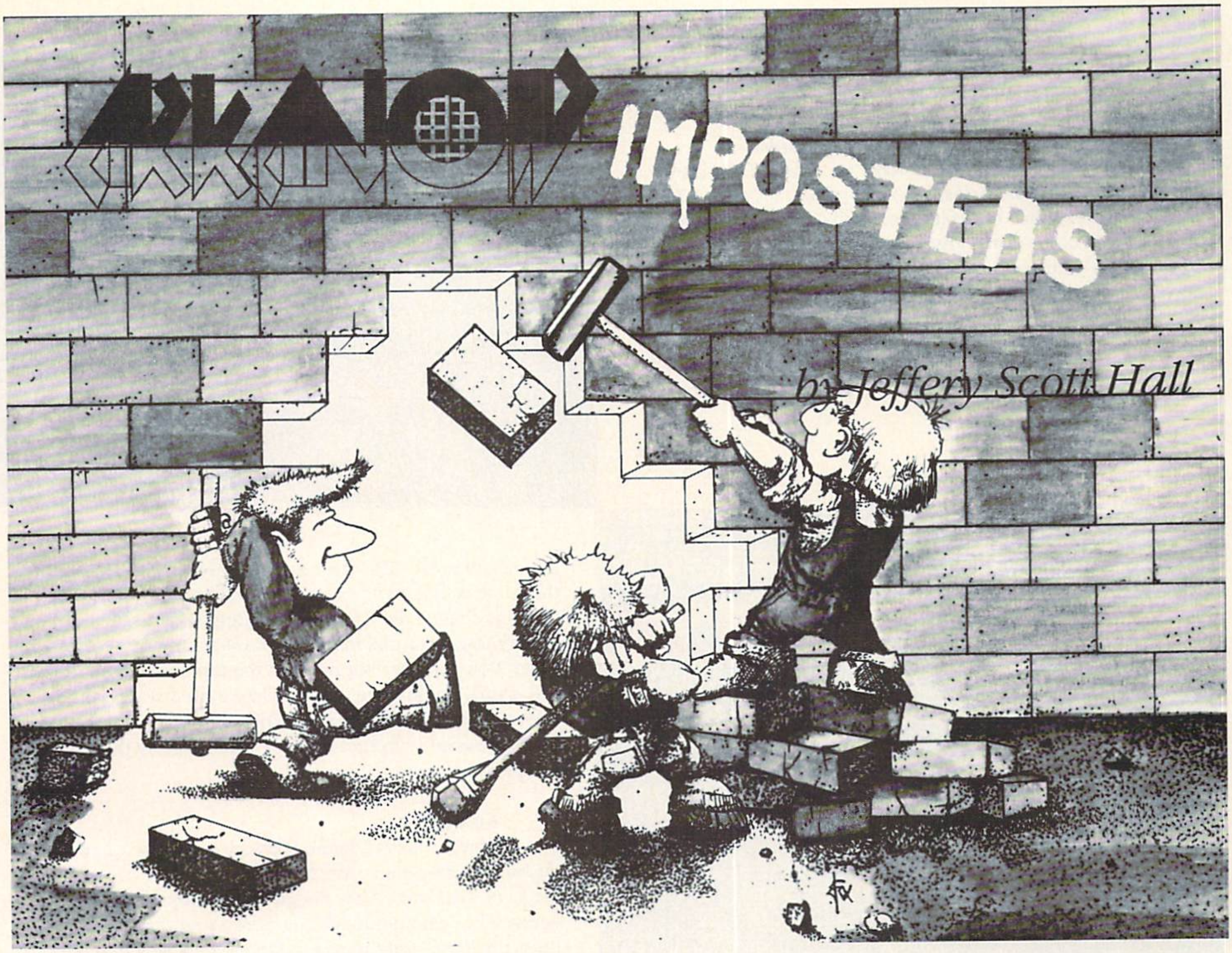
Pageflipper Plus FX
$159.95
Mindware International
110 Dunlop St. West
Box 22158
Barrie, Ontario
Canada L4M 5R3
(705) 737-5998
Orders (800) 461-5441

Animator:Flipper
$59.95
Hash Enterprises
14201 SE 16th Circle
Vancouver, WA 98684
(206) 256-8567

Backlight
$49.95
Silent Software
706 W. Broadway Suite 202
Glendale, CA 91204
(818) 243-4540

Zoetrope
$49.95
Antic Software
544 2nd. Street
San Francisco, CA 94817
(800) 234-7001

Fantavision
$59.95
Broderbund Software
17 Paul Drive
San Rafael, CA 94903
(800) 527-6263



ARKANOID IMPOSTERS

by Jeffery Scott Hall

Since Discovery Software released Arkanoid, four games similar to this classical game— each with a different twist of their own—have been released. Now that you have greater variety , it will be more difficult to choose among them. This feature will describe all four Arkanoid-type games so that you know their differences. Before we start looking at the four different games, let's answer one simple question.

What is Arkanoid?

If you have never heard of Arkanoid before, you must have been stationed on another planet. (Or you're a college student like me; which is close to the aforementioned). But to answer the question, Arkanoid is a reincarnated brick and paddle game of the old classic with some major enhancements. The object of Arkanoid and its "imposters" is to hit a bouncing ball with your paddle to knock out every brick in the wall. The mouse controls the paddle, and the ball's speed depends on the row of bricks you hit. And as if that weren't enough, moving obstacles will change the ball's current direction. Other enhancements to the original include different backgrounds, brick patterns, music, sound effects, and even bright colors. But without any further delay, let's explore the first of the five games.

Arkanoid

From out of the arcade and into the home comes the #1 1986 smash game Arkanoid. This is the first and only game licensed direct from Taito by Discovery Software. When the game is loaded, you will be in demo mode. Arkanoid provides you with a choice of either one or two player competition. When the game begins, the spaceship Arkanoid is under attack, while the shuttle craft Vaus is escaping. Your plans of a safe departure are foiled by a sinister alien who has trapped you in a space warp—a dimension in which objects are formless. Choose your starting level by rolling the mouse left or right. Levels range from 1 to 20. Even though there are actually 66 levels of play, the highest level you may start at is 20.

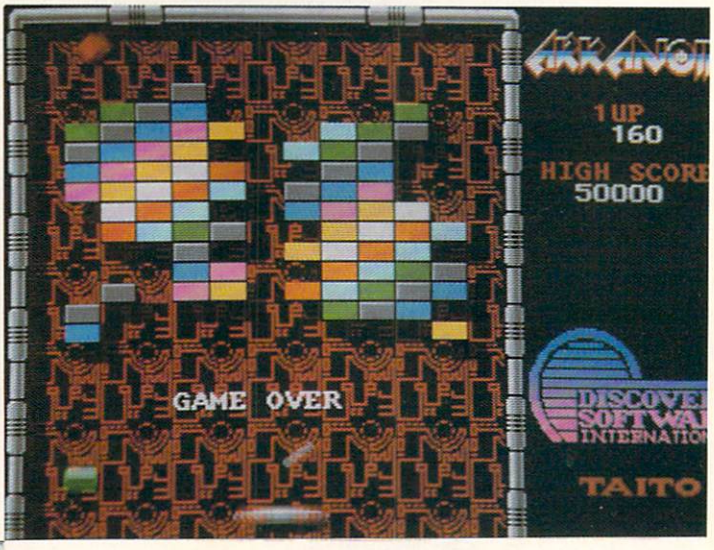
When you move your mouse, you'll see each game level exactly as it appears when you're on that level. Each level is unique and provides you with a variety of scenery, ranging from a space invaders-type alien to the sinister evil alien himself. When you choose your starting level, click the left mouse button to start play. The Vaus will be at the bottom of the screen along with the energy ball (the little thing that bounces around the screen). To release the energy ball, either click the left mouse button or wait for the automatic timed release. It is your job to shatter the space

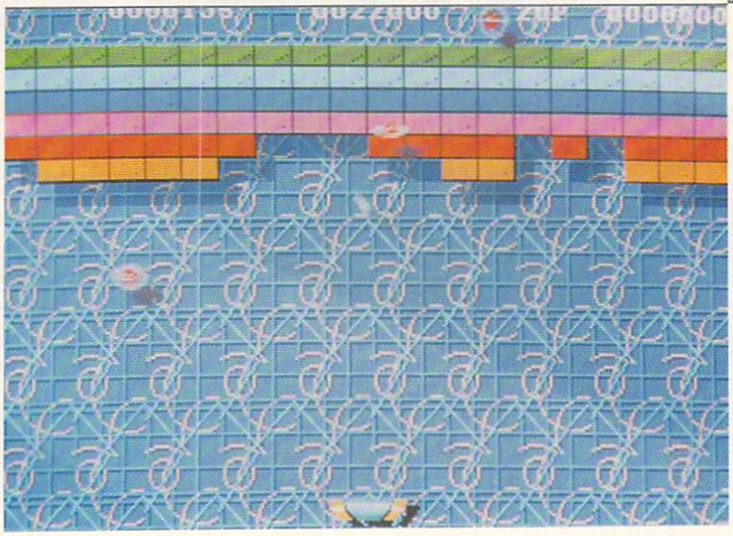
(continued)

walls and defeat the enemy forces, which is no simple task. The energy ball's speed depends on the color of the wall; and the faster the ball moves the faster your score moves.

Opopo, do I *have* to take my power pill?

Sound too easy for you arcade "wizards"? Then how about adding some obstacles, such as Pyradoks, Opopos, Tri-Spheres, and Konerds to make it more challenging? These obstacles will come down toward you, hover above your ship, try to deflect your energy ball, and make you wish you had never seen them. The higher the level, the more advanced the obstacle. Each time you defeat the space walls and obstacles, or break into the next level with a power capsule, the levels get tougher. You'll need the reflexes of a Sunday driver on a downtown expressway!





At times, you'll break through the bricks to find "power capsules" drifting down toward you. These help re-enforce the Vaus, and they give your ship an extra function. There are seven different capsules in all: catch (allows you to catch the energy ball); slow (slows down the energy ball); expand (enlarges the Vaus); disruption (splits the energy ball into three); laser (enables your ship to fire lasers); break (allows you to break in the next level without completing the one you are currently in); and paddle (gives you an extra ship).

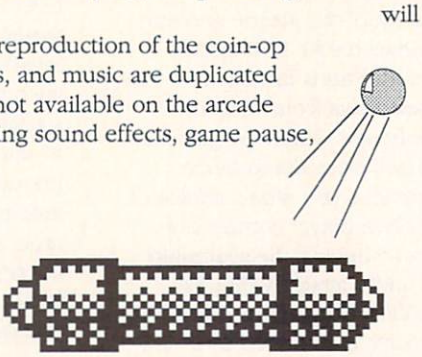
Arkanoid is an almost exact reproduction of the coin-op version. The graphics, sound effects, and music are duplicated almost exactly, as are a few extras not available on the arcade version. These include on/off toggling sound effects, game pause, and resume play. This last one lets you keep advancing after the game is over, with your score will be set to zero. You can also change the speed of the energy ball from very slow to very fast. Arkanoid loads into RAM and never accesses the drive. You can put your name in the top five bravest players if you qualify; but your score will be erased when you reboot or turn the computer off. This is the only minor inconvenience I have found, so I can safely say that Arkanoid is true to the original.

Amegas

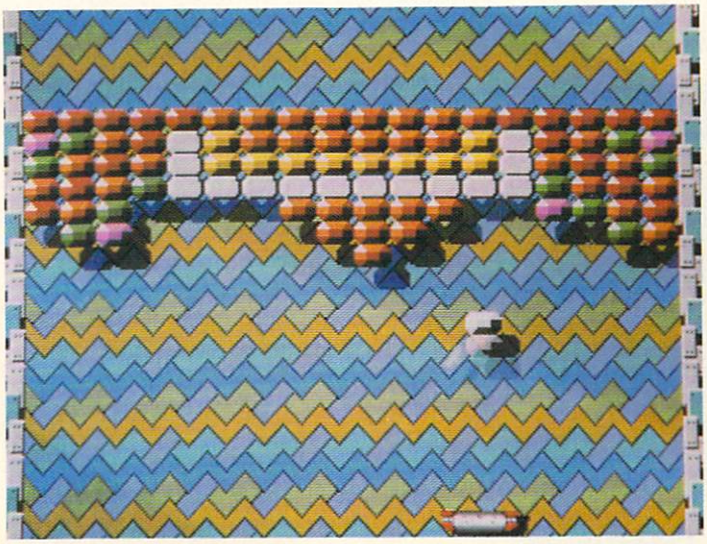
Amegas, from Digitek Software, takes the same ingredients Arkanoid uses, but lightly sprinkles on some magic of their own. After the game is loaded, you will find yourself in demo mode while a stereo soundtrack plays. Press the left or right mouse button to choose a one or two player game. Your object is to shatter through 40 different levels of play.

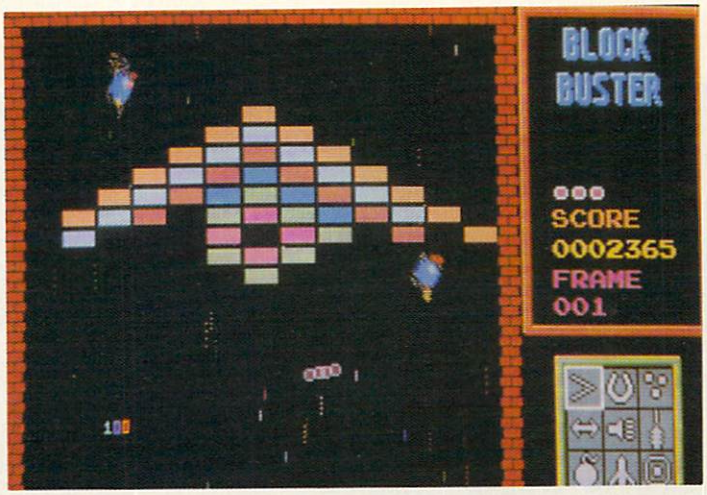
While this game remains close to Arkanoid, you will find some very unique differences. For one, you must start at level one and work your way up. Also, when advancing from level to level you will find that the background scenery changes along with the bricks. For example, on level two you will see only four bricks until the last one is hit, then it will explode revealing more bricks in a different pattern. The Amegas Stones (i.e. power capsules) are the same, but the colors provide the paddle with an enhanced function. And while the obstacles act the same as those in Arkanoid, you will find that they're different in appearance and variety.

You can't pause during game palay. However, when you get killed or start a new level, you will be prompted to press the left mouse button to continue, thus allowing you to take a break if you want. There



is no sound toggle. This can be very annoying when you make it to the 31st level and hear the pong for the thousandth time. Whenever the game is over, you will be prompted to press the right mouse button, if you wish to continue play from where you left off, if you not, press the left mouse button to start over. If your score is among the top ten, then you will be prompted to enter your initials, which will be stored on disk. The different graphics, sounds, and music combine to make it unique while providing you with the same rules of Arkanoid.





Blockbusters

Blockbusters, from Mindscape Software, certainly provides a different approach to the Arkanoid games. The object is to use your paddle break through all the bricks in the jam-packed 80 levels of play. One difference is that you can create up to 48 screens of your own, which combines with the original 80 for a grand total of 128 different brick patterns! It is impossible to describe how this is done in the space requirements of this feature, but I will assure you that it is very easy to learn.

The background scenery doesn't change. In fact, there isn't a background at all. Instead, the different brick patterns will be displayed on a vertically scrolling star background. The "tokens" (i.e. power capsules) are the same as the others with the exception of the following four. Torch lights up invisible bricks; smart bomb kills all aliens until the next level; missile destroys multi-hit bricks with a single hit; and force field encircles the ball and allows it to pass through aliens and bricks without any deflection. To choose a weapon, press the right mouse button until the weapon you want is highlighted.

Deposit another token for the next weapon, please.

Each weapon requires a certain amount of tokens. Force field, for example, requires 9 token. The floating obstacles react the same way as that in the other games and appear as only two different shapes. Another important feature is the password access on the title screen which appears before game play.

After you complete a set of 10 screens, you will be given a password for future use. Depending upon the password you enter on the title screen, you will be taken to a more advanced level, allowing you to skip the easy ones you have mastered. If your scoring is high enough to be in the top ten Blockbusters, you will be prompted to enter your name in order to save it to disk. It does include a pause option during game play, which is nice for a needed break.

This is definitely the most unique game, of all the "Arkanoid Imposters" described here. However, I feel that this game lacks in one specific area: graphics. The graphics are comparable to that of Blockbusters on the Commodore 64, and is a noticeable port from that computer. Even though I detect a slight enhancement over the graphics, I feel they could have done a better job. Despite this, I have found the sound to be excellent, and the ability to design your own brick patterns is a plus. If you are looking for a different version, look no further.

Crystal Hammer

Have you ever wanted to break crystal in outer space? Well, it might not be exactly number one on my priority list, but on the game list it's a different story. Your object, as you mayb have guessed, is to smash through 30 levels of crystal. How is this done? With a hammer, of course. There are eight special pieces of crystal (i.e. power capsules) that when caught, gives your hammer a special weapon.

The background and floating obstacles are wacky and funny. Crystal Hammer gives you some extremely beautiful backgrounds along with some well known characters, including wiggle worms or a pacman chomping across your screen, as well as some other

(continued on page 46)

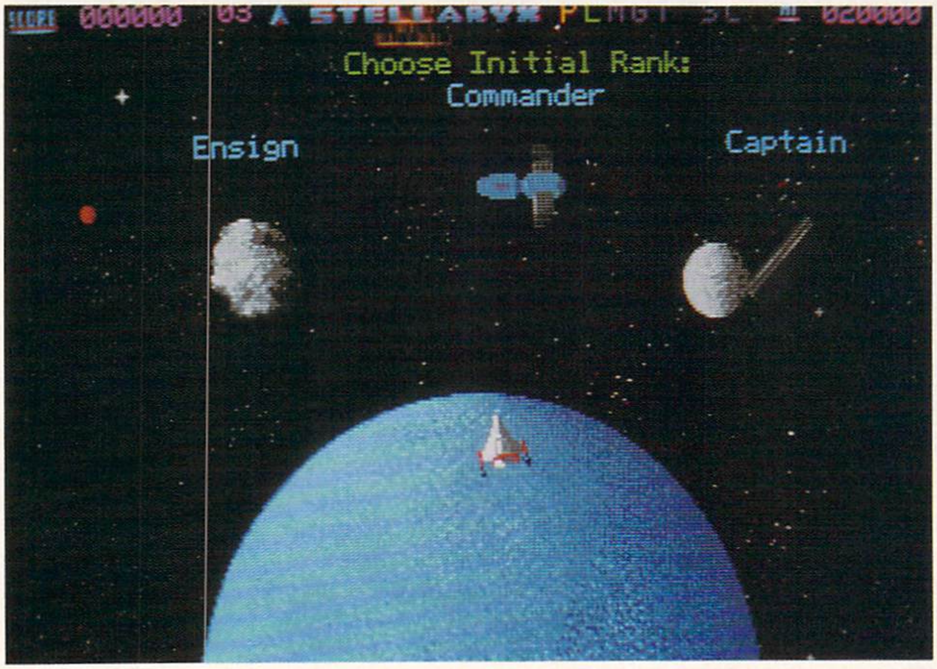
STELLARYX

by Stephen Kemp

Remember Asteroids, the arcade game? Perhaps you found it challenging and spent hours blasting those rocks. Or maybe you grew tired of it because there wasn't enough variety. Well, no matter how you felt about the original, something better has come along. It's STELLARYX, from Laser Gamesmanship!

Asteroids with a twist

STELLARYX is a new game that is much more exciting and challenging than Asteroids ever was. Of course, you are expected to shoot rocks and aliens, but there are also other tasks to perform. Blasted rocks can reveal beneficial equipment that your ship can use or might reveal aliens. You can collect lasers, guided missiles, stasis grenades, tractor beams, shields, cloaking devices, and astronauts. All of these things can make your job easier—if you know when and how to use them effectively.



The ship can be controlled by either the mouse or the joystick. However, I found that using the joystick was rather difficult. Players can select from three different skill levels when the game begins. Once a screen has been cleared of rocks, aliens, and other space debris, you advance to the next screen to face a new (more difficult) assortment of space "things". For those with at least 1 meg of memory, you occasionally get to earn a bonus by performing a docking maneuver at a space station.

This game also has something not usually found in a game—HELP. That's right, just press the Help key and the game pauses and lets you review the online help. The help contains information about the game controls, the enemies, weapons, and strategies to use during play. Help is another fine element of this game, but even if you don't need it, it is sometimes convenient to pause the game just to take a break.

Color copy protection

The only unsatisfying thing about this game is the copy protection scheme. Although many people have very strong feelings against copy protection, some of us are a little more tolerant. But unlike other programs that have hard copy protection or require you to type in a word from the manual, this one uses a color chart. The color chart proves more effective than the "key word" method because it can't be xeroxed. Unfortunately though, it is sometimes hard to enter the correct codes because of variations between the color shades on the screen and those on the chart.

Overall

It is very easy to spend hours playing this game. The graphics are very good and the game contains terrific stereo sound effects. Initially, it may seem that it is not easy to play this game, but after a few minutes the addiction takes over. Asteroids move over, STELLARYX is here!

•AC•

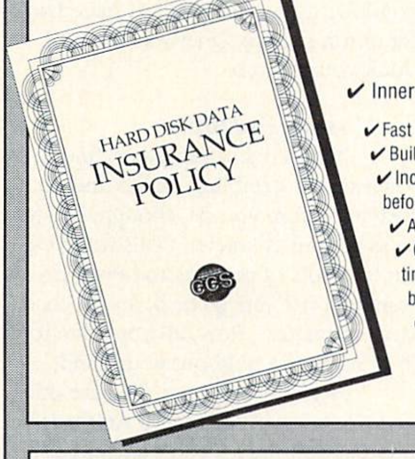
STELLARYX
Laser Gamesmanship
P.O. Box 1223
Alta Loma, CA 91701-8223

FOR THE AMIGA

Protecting your valuable hard disk files is easier and faster than ever before!

QUARTERBACK™

*The **FASTEST** Hard Disk Backup Utility!*



Backup to or restore from:

✔ Floppy Disks ✔ Streaming tape (AmigaDOS-compatible)
✔ CLtd's Konica 10.7MB high-density floppy drive
✔ Inner-Connection's Bernoulli drive ✔ ANY AmigaDOS-compatible devise

✔ Fast backup - 20MB in 30 minutes or less ✔ Uses two floppy drives (if available) for backup/restore with automatic switching ✔ Builds, sorts and displays catalog of files and subdirectories ✔ Provides FULL/Subdirectory/Individual file backup/restore ✔ Includes or excludes files by name (with wild cards), file date, or archive bit ✔ Calculates the number of floppies you'll need before you start ✔ Handles files of unlimited length, unlimited subdirectories and unlimited files per subdirectory ✔ Automatically formats diskettes with no delay as it writes ✔ Sequentially numbers and date/time stamps backup diskettes ✔ Checks the sequence number and date/time stamp of each diskette before restoring files from it ✔ Restores original date/time stamp, file notes, and protection bits on both files and subdirectories ✔ Runs with Workbench or CLI ✔ Produces backup/restore report to disk or printer ✔ Beeps for floppy change ✔ Accepts CLI parameters and batch command files ✔ Detects bad disks during backup or restore ✔ Convenient/user friendly error recovery ✔ Multitasking ✔ Runs in 512K ✔ No copy protection ✔ Works with all AmigaDOS compatible hard disk drives.

Only $69.95 Plus $3.00 shipping and handling, CA residents ass 6% sales tax.

DISK 2 DISK™

Convert C64/C128 Files to the Amiga!

DISK-2-DISK™ makes it easy and convenient to transfer C64/C128 files to and from the Amiga! DISK-2-DISK programs the Amiga model 1020 external 5.25" disk drive to read and write 1541/4040 and 1570/1571 disk formats including 1541 "flippies".

✔ Converts Commodore/PET ASCII to AmigaDOS standard ASCII and vice versa ✔ Transfers word processing text files (such as PaperClip, SpeedScript and Pocket Writer) to and from the Amiga for use with popular Amiga word processors ✔ Includes 3 public domain programs for converting C64 Koala, PrintShop and Doodle files to IFF format ✔ Finds and flags dialect differences between Commodore Basic and Amiga Basic files ✔ Provides VALIDATE BAM and CHECK DISK utilities (VALIDATE BAM verifies the directory structure of the 1541/1571 diskette; CHECK DISK reads every block of a 1541/1571 diskette to detect diskette errors).

DISK-2-DISK requires the Amiga model 1020 5.25" disk drive.

Only $49.95
Plus $3.00 shipping and handling
CA residents add 6% sales tax.

DOS 2 DOS™

DOS-2-DOS transfers MS-DOS and Atari ST files to and from AmigaDOS!

DOS-2-DOS version 3.0 permits access to any MS-DOS volume available via AmigaDOS, including MS-DOS partitions on hard disks and MS-DOS volumes on LANS or SCSI networks.

✔ Supports single and double sided 5.25-inch as well as 3.5-inch 720KB MS-DOS diskettes ✔ Reads/Writes 3.5-inch Atari ST diskettes (GEM format) ✔ Reads a variety of 5.25-inch MS-DOS floppy formats via the CLTD Konica high-density floppy drive ✔ Converts ASCII file line-ending characters and provides Wordstar compatibility ✔ Supports full directory path names, with wild cards in the file names ✔ Allows selection of MS-DOS and AmigaDOS subdirectory and displays sorted directory listing ✔ Formats 3.5-inch and 5.25-inch MS-DOS diskettes and Atari ST diskettes ✔ Provides duplicate file name detection with query/replace options ✔ Provides TYPE and DELETE commands ✔ Permits renaming of files where file name restrictions occur ✔ Remains resident to permit AmigaDOS disk swapping.

Only $55.00
Plus $3.00 shipping and handling
CA residents add 6% sales tax

CCS

Central Coast Software™

268 Bowie Drive, Los Osos, CA 93402 • Telephone: 805 / 528-4906 • FAX: 805 / 528-3138
DEALER INQUIRES WELCOME

MasterCard VISA

A look at... Questron II

by Jeffery Scott Hall

A journey back in time and space

Mesron, the great wizard, has sent you back through time and space in order for you to stop the Evil Book of Magic from being created by six mad sorcerers. It is your job to find them and prevent the consummation of their abominable deed. You must explore Landor, gaining knowledge as well as weapons for use in your quest.

Questron II, from Strategic Simulations Inc., sends you back in time to destroy the Evil Book of Magic. Once the game has loaded, you will find yourself looking into the book of Questron. From here you have three options: Start a new game, Demo, and Instructions. Once you have decided to embark upon your journey, the fun and danger begins!

While exploring Landor, you will encounter dungeons, tombs, cathedrals, and castles, each posing a different threat. While you explore these underground dwellings, a map at far right will show you where you have and have not been. But, once you leave the underground dwelling, the map is lost forever. The view you have inside one of these underground dwellings is a solid 3D perspective of both your surroundings and your enemies. It must be seen to be appreciated.

There are 60 different monsters in Landor. Some are very friendly while others are vicious. Some monsters you encounter will be strong, others will be weak. When you defeat a monster, you may find special items such as weapons, gold, food, and armor in its place.

On the far left of the screen you will find a listing of the options available during game play, like magic, inventory, etc. To activate one of these items, click the left mouse button over it. Either the appropriate action will occur or you will be presented with a requestor from which you may choose other options, use magic, cast fireball, etc. Just below the options, you will find your character's current status displaying his hit points, food, and gold.

To explore Landor, you must either walk or run through it by either clicking the left mouse button or moving the joystick. Your character will react accordingly, and move in the direction you choose. The screens scroll constantly, even during underground exploration! And during your journey, you will find many different towns to enter which will help you in your quest.

A stranger in town.

In town, you will meet many friendly and not so friendly characters. You can find out a lot about the town and the surrounding area if you approach these characters in the right way. In the town you will find areas to eat like a barbarian, sleep, cheat at gambling, and buy weapons and armor—your average adventurer's night out. If you play your cards right, you'll come out of town rested, full of food, heavily armed, and a few thousand gold pieces richer. On the other hand, if you get a bad reputation, you might find yourself six feet under!

Summary

I have been playing Questron II for quite some time now and it seems like every time I play I see something new. It requires a fairly long time to get your character built up, but once you do you'll have more fun playing. I only two minor complaints: there's no pause option; and you can't save an underground map. It would have been very nice if to have a save routine to recall the map of an underground dwelling you've already explored. I found the lack of a pause option to be very inconvenient when I needed a break from all the hack and slash. However, these are just minor complaints, and overall I found Questron II very enjoyable.

Questron II is a very exciting dungeons and dragons type game with great graphics and sound effects. You will find an option which will allow you to save your character to disk so you can continue if you get killed. I have had a lot of fun playing Questron II, and I think you will too!

Hints and tips

When first starting out, find a town with a gambling casino and bet heavy. Before you do, though, be sure to save your character. This way you can win lots of money; and even if you lose, you can just go back and re-load your character. Now, all you have to do is sit back and hold out your hand!

If you have armor, be sure that you don't forget to wear it. Armor is no good to you unless you put it on. But the heavier the armor, the more strength you'll need to wear it. If you find heavy armor when first starting out, stay with the leather until you have built yourself up.

While attacking high level monsters, know when to call out the heavy stuff. If you're getting kicked from one end of the map to the other, check your book of spells and start casting. However, if you have bitten off more than you can chew, it is safer to cut your losses (unless you back up your character).

Make sure that you talk to the people inside towns and villages. They might have very useful information which could come in handy later in the game. It is best not to pick a fight with any of the town's people if you can help it, for they might have some pretty mean friends that don't take kindly to strangers.

•AC•

[*Look for a full-length review in an upcoming AC.—Ed. C^3*]

Questron II $49.95
Strategic Simulations, Inc.
1046 N. Rengstorff Ave.
Mountain View, CA 94043
415-964-1353

Fox

Attention AMIGA Users!

GEN

LOCK

GEN/ONE™: Professional Quality Genlocking For All Amiga Computers
Specially designed for compatibility with the Amiga® line of computers, GEN/ONE is the premier genlocking encoder.
If you're serious about your video graphics, this interactive desktop accessory is a no-nonsense component. And to back our commitment to quality and customer support, we give the best warranty in the business.
GEN/ONE from CSI...Quality. With the commitment to match.℠

GEN/ONE's Advanced Features Include:
USER CONTROL
- Separate RGB and Video Gain
- H, V, Subcarrier Computer Timing
- Computer or Video Sync Source Selection
- Separate Y/C and Video Gain
- 3 Modes of Operation

SPECIAL FEATURES
- Y/C Output for Super-VHS
- Comb Filtering In Luminance Channel
- Separate Power Supply
- Convenient RGB Computer Output

$895.00
with 1 year warranty

Call For More Information
And A Free Brochure
(516) 273-0404

Communications Specialties, Inc.
89A Cabot Court
Hauppauge, NY 11788

Amiga is a registered trademark of Commodore-Amiga, Inc. GEN/ONE is a trademark of Communications Specialties Inc.

(*Arkanoid*, *continued from page 41*)

strange faces known to us arcade addicts. If you are lucky enough to shatter enough crystal, you will be prompted to enter your initials for the top ten, which will be saved to disk.

I have three minor complaints with the game: no sound toggle, only thirty levels of play, and no pause feature. This does not detract from game play at all. The faultless combination of stunning graphics, sound, and overall appearance make this the most wonderful of all the "Arkanoid Imposters."

Decisions, decisions.

You might find all the different "Arkanoid Imposters" confusing, and you might want need of some advice about which one is the best, right? Allowing for personal taste, I can make two suggestions which might interest you. First, if you are looking for an almost exact Arkanoid reproduction, go with Arkanoid from Discovery Software. Which "Arkanoid Imposter" to choose? Well, my two personal favorites are Amegas and Crystal Hammer, which provide the most unique twists of all. However, I can safely say that no matter which one(s) you select, none of the above will be a disappointment.

The Arkanoid saga concludes...Or does it?

Whew! If you've managed to make it to the end of this feature you deserve a medal (or even better, some free tokens for different games at the local arcade)! Hold on to your eyeballs for just a little longer, we're almost done.

In the latest news bulletin from somewhere deep within Arkanoid space, there is word that another Arkanoid game is in the works. Yes, Taito themselves have decided to market their own original version of Arkanoid which is due out sometime this winter, and thus the saga continues. Now, if I can just decide which version to play, any suggestions?

•AC•

Arkanoid
Discovery Software International
163 Conduit Street
Annapolis, MD 21401
1 (800) 34-AMIGA
$49.95

Amegas
DigiTek, Inc.
10415 N. Florida Ave
Suite 410
Tampa, FL 33612
(813) 933-8023
$34.95

Crystal Hammer
Constellation Software
17 Saint Mary's Court
Brookline, MA 02146
(617) 731-8187
$19.95

BLOCKBUSTER
Mindscape, Inc.
3444 Dundee Rd.
North Brook, IL 60062
(312) 480-7667
$39.95

Las Vegas COMDEX Report

by Louise Brinkmann

Another COMDEX was at hand, and there was a feeling of anticipation and excitement in the crowds waiting to get into the show. Commodore's booth in the West Hall was outstanding—its wall of monitors displayed the "Only the Amiga Makes It Possible" video and the Amiga commercials currently airing on MTV. These commercials, created entirely on the Amiga, are entertaining and informative.

It was sometimes difficult to get around the Commodore booth because of all the people—the Amiga still attracts a lot of attention at COMDEX! Commodore announced a number of new products at the show, including the Amiga 2500 computer. The Amiga 2500 is basically an Amiga 2000 with a 68020 chip and standard 3 megabyte of memory which is expandable to 9 megabytes. The computer will be available soon, at a $5,000.00 list price.

Also announced was the Amiga 2500 UX, an Amiga 2500 with the multi-tasking, multi-processing environment called AMIX, which is based on UNIX System V. This workstation is an outgrowth of the Amiga 2000 system with addded features like fast mass storage, memory management, high speed processing and integral tape backup.

An Amiga with a Transputer board was also announced and demonstrated. The Transputer is a board containing a parallel processor which allows it to run simultaneously with the Amiga's CPU. So in addition to multi-tasking, you also have multi-processing. The Helios operating system, included with the Transputer board, was designed to be multi-tasking, multi-user, and multi-processing. This operating system uses all of the available processors and coordinates their activities. All of this is integrated into the Amiga and is transparent to the user and the program being run. The Transputer board and

(continued)

Helios operating system are not yet available in the United States.

New to the Amiga 500 line is the A590, a 20 megabyte hard disk with 2 megabytes of RAM. It also has a SCSI interface so you can easily add hard drives or CD ROM units. The A590 is very compact and fits on the left hand side of the Amiga 500. I was impressed by its speed.

The area of the booth that really drew the crowd was the demonstration of the high resolution color graphics card. It permits faster color bit map processing with graphics displays of up to 1 million pixels. It also has programmable resolutions of up to 1024 x 1024 pixels, with 8 image bit planes providing 256 colors, plus 2 overlay bit-planes providing an additional 3 colors simultaneously. The graphics and animation sequences generated with this board were outstanding. As of this writing, no release date or pricing had been announced.

SpectraScan color scanner

Perry Kivolowitz from ASDG used a JX-300 scanner to demonstrate SpectraScan, ASDG's professional color image scanning system. The interface is very easy to use and allows varying resolutions of 30 - 300 dpi with increments of 1/100th of a dpi. SpectraScan software allows four color separations, color adjustments, and resizing the scanned image. All operations are in 24 bits, and you can save in any format such as IFF and HAM. I was pleasantly surprised with the speed of this scanning system. The official release date is January 1989. The JX-300 scanner will list for $4995.00, and the SpectraScan software will cost $995.00. Scanned images were printed out on Howtek's Pixelmaster printer, a wax printer which prints using melted wax instead of ink. The results are simply marvelous.

DeluxePaint III to Offer Animation

Dale Weideman of Electronic Arts demonstrated DeluxePaint III, which adds animation capabilities to their superb paint program. Finally animation made easy! Some of the improved features included in the painting portion of the program are extra halfbrite, 64 color mode, overscan painting, new fill modes, more extensive air brush capabilities, and ability to use fonts from any drawer or disk. The speed is now done in real-time with no delays.

The animation included in DeluxePaint III makes animation as easy as painting. The demonstration I saw was fantastic. Dale started out with a painting of a waterfall and a river, then animated a boat moving along the river. Finally, he did another animation of a flag waving and attached it to the boat. So, DeluxePaint III allows you to have animation on top of animation—a neat feature.

If you are interested in animation, I suggest you take a look at this package. It's due during the first quarter of 1989, at $129.95 list. DeluxePaint III runs on any Amiga system with one megabyte of memory, and is compatible with all Amiga products that work with DeluxePaint I and II. Owners of DeluxePaint I or II who purchased the product before Dec 1, 1988, can upgrade by sending in the manual cover and a check for $50.00 plus $7.00 for shipping and handling. If you purchase DeluxePaint II after Dec 1, 1988, send the dated receipt and $7.00 for shipping and handling.

Sculpt-Animate 4D

Speaking of animation, BYTE by BYTE's Scott Peterson was demonstrating Sculpt-Animate 4D, which is similar to the original Sculpt-Animate with added features for power users. New features include interactive object editor, enhanced file requestors, assignable hot keys, significantly faster ray tracing, improved anti-aliasing, and animation scripting language. Sculpt-Animate 3D users can upgrade for only $195. Sculpt-Animate 4D lists for $499.95 and is currently shipping.

Copper Bittner from the Zuma Group displayed TV Text and Zuma Fonts. TV Text allows you to create professional quality lettering for screen shots, videos, or hard copy printouts. I found it an easy to use and invaluable program. TV Text is available now for $99.95. Zuma Fonts volumes I, II, and III are fonts packages featuring range 12 to over 100 point sizes. These font packages list for $34.95

I expected to see WordPerfect demonstrated in Commodore's booth; instead Jack Appleman from Micro-Systems Software showed Excellence!, a "what you see is what you get" Amiga word processor. Features include indexing, spell checking, thesaurus, grammar checking, graphics, and multiple column capability. It is currently available and lists at $299.95.

Notes From Aegis

Scott Oransky and David Hodges from Aegis showed Sonix and AudioMaster II. Sonix is a music program that allows you to compose your own music by pointing and clicking with the mouse. The interactive sound controls allow you to change the tempo, transpose, or tune your music as it is playing. AudioMaster II is a digital sampling and editing program that loads a digitizing sound into the Amiga and transforms it into a totally new and unusual sound just by cutting and pasting different segments of that sound. Just think of the sound effects that you can create! The retail prices for Sonix and AudioMaster II are $79.95 and $99.95, respectively.

Burton Robson from Gold Disk demonstrated Professional Draw v1.0, a drawing program that creates high-quality professional illustrations and drawings which can be printed on a dot matrix printer, PostScript compatible printer, or color PostScript compatible printer. The output generated on the QMS ColorScript 100 (color PostScript) printer was very impressive! Professional Draw is very powerful but very easy to use. It offers a variety of geometric and freehand drawing tools, access to over a million colors, full text and color separation capabilities. Professional Draw lists for $199.95 and will be released in December 1988.

Digi-Paint and Digi-View were shown by Laura (Maxine Headroom) Longfellow from NewTek. This combination of digitizer and HAM paint program is still one of the best buys in the Amiga market. Digi-Paint offers an impressive array of colors and drawing tools. With just the flick of the mouse, you can change the color of a specific area in your HAM (digitized) picture.

One thing I noticed was that easy to use but powerful programs are becoming the norm rather than the exception in the Amiga marketplace. All in all, COMDEX was an enjoyable experience.

•AC•

WIN A MONITERM!

Build an expert system application with the MAGELLAN™ artificial intelligence expert system building tool and win a MONITERM Viking I high resolution monitor for your AMIGA.
Imagine the possibilities: the power of the AMIGA coupled with the flexibility of MAGELLAN!

Categories for the First Annual MAGELLAN Application Contest are:

Most Productive Application
Business or Diagnostic systems, decision assistance tools for financial investment, production control systems, or inventory management systems.

Most Creative Application.
Music or Animation systems, or innovative approaches to traditional expert systems.

Most Bizarre Application.
Let your imagination run wild!

3 Grand Prize Winners will receive:
A Moniterm Monitor
A MAGELLAN Developer Package
Retail Value $3000.

2 Runners-Up in each category will receive:
MAGELLAN Developer Package
Retail Value $1000.

All finalists will also receive a MAGELLAN designer sweatshirt.

All entries must be received by Emerald Intelligence by June 1, 1989. Contest void where prohibited. Contest Winners will be announced at Ami-Expo Chicago in July 1989. All entries become the exclusive property of Emerald Intelligence Inc., and cannot be returned. Winning entries may be used in a promotional campaign and any entry may be included in promotional materials or application manuals. For a complete set of rules and an entry form, send a self addressed stamped envelope to MAGELLAN Contest Rules, c/o Emerald Intelligence.

MAGELLAN™

THE EXPERT SYSTEM SOFTWARE

Emerald Intelligence

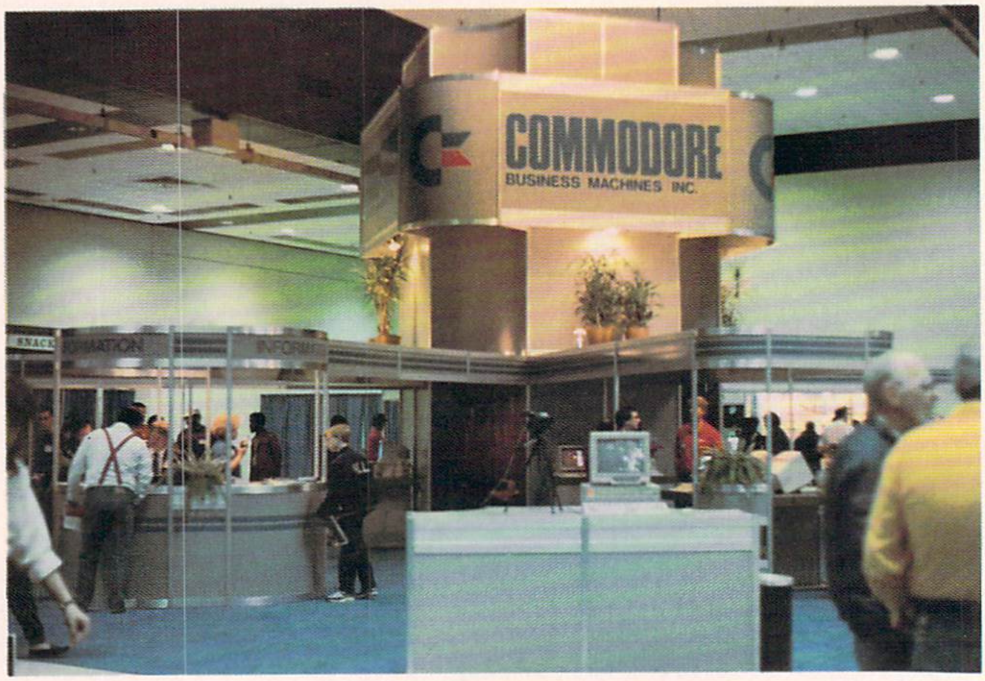
334 South State Street, Ann Arbor, Michigan 48104 (313) 663-8757

MAGELLAN is a trademark of Emerald Intelligence Inc. AMIGA is a registered trademark of Commodore-Amiga Inc. Moniterm is a trademark of Moniterm Corporation.

Philadelphia World of Commodore

by Chris Darsch
Rick Rae

The first U.S. World of Commodore show was held at the Philadelphia Civic Center from November 3-6. A long-standing fixture in Canada, this was the first time Commodore spomsored the show in the United States. Representatives from organizers The Hunter Group and Commodore said they were pleased with attendance, which fell near the lower end of the expected range of fifteen to twenty-five thousand people.



Thursday and Friday were exceptionally long days, running from 10 AM until 9 PM. Each evening's announcement that the show was closing was greeted with a spontaneous burst of applause from the weary exhibitors. Crowds were heaviest on Saturday; Eric Miller of Comp-U-Save—a company which appears at upwards of 75 shows a year—said was the biggest sales day Compu-Save ever had.

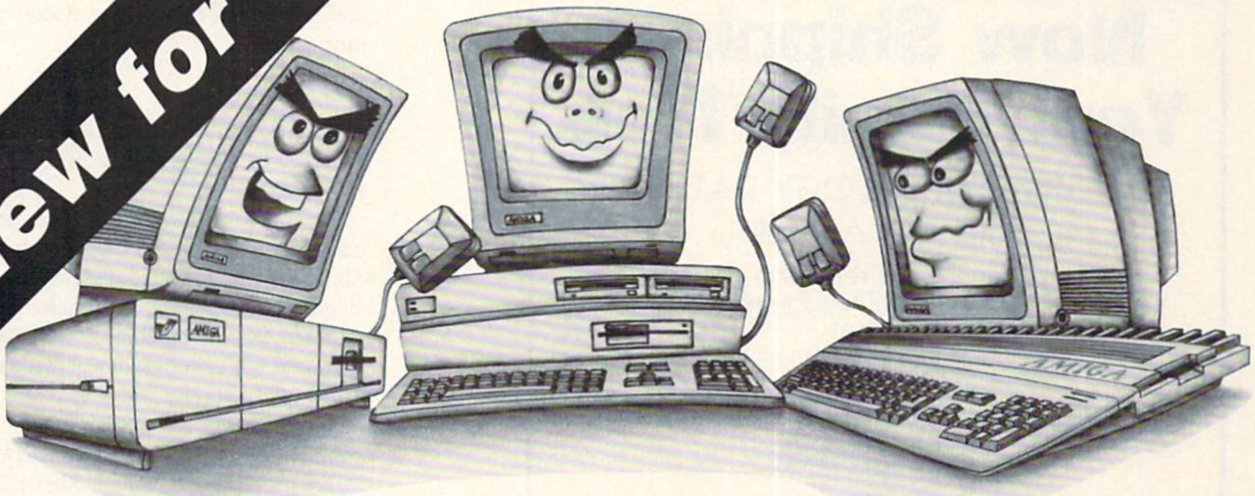
The show featured over 75 companies—ranging from Commodore and its mammoth "media center" to one-man operations and product representatives working under other companies' banners. The vast majority of the exhibitors showed Amiga-oriented products and services, whoch caused some C64 and C128 owners to express disappointment at the lack of representation for their machines. This trend will no doubt continue. In fact, many Amiga-specific companies who didn't have booths indicated they would be back in force next year—once they saw the emphasis placed on the Amiga.

"We insist on professional-looking convention space and exhibitor displays, but we have no aspirations to be a Comdex," said Gordon Hunter, president of The Hunter Group. "We want to keep a fairly small show, one that caters to the average Amiga owner and potential owner, where someone can not only see the products but purchase them from one of the retailers, " he added. Mr. Hunter, who makes a point of attending each show his group produces, was pleased with the turnout. "Attendance figures are comparable to the first Toronto show six years ago, " Hunter said. " And this year, the Toronto show will be three times the size of this show. So, we will grow in the U.S. as we have in Toronto, slowly and carefully."

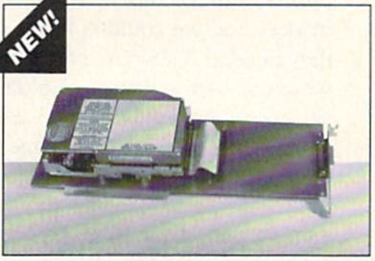
(continued)

POWER HUNGRY

New for '89!



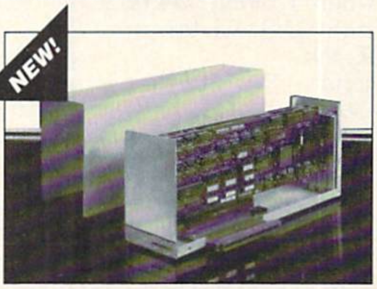
Whether you own an Amiga 500, 1000 or 2000 you'll want to feed it right. Fortunately we're here to help. At Expansion Technologies we provide high quality expansion products that are guaranteed to keep your Amiga growing happy and strong. Just check out our menu :



FLASH! CARD *A2000*
Hard Disk Interface
DMA/SCSI/AUTOBOOT

The Flash! Card/2000 is a new SCSI/DMA hard drive interface for the Amiga A2000. The Flash! Card takes full advantage of the improved features available under AmigaDOS 1.3, including sockets that allow your Amiga to boot directly from the hard drive. The Flash! Card also features ram cache DMA circuitry providing up to 4 megabytes per second transfer rates (Under the Fast File System transfer rates are even faster than the drive mechanism). The Flash! Card allows direct low-profile mounting of readily available 3.5 inch hard drive machanisms or the interface can be connected to the internal drive. The DB25 connector allows the use of external drives or other SCSI devices.

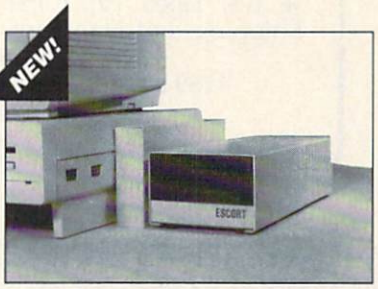
Flash! Card/2000	**$229.95**
Flash! Card/32	**$749.95**
Flash! Card/48	**$849.95**



TOOL BOX *A500/1000*
A2000 2 Slot Expansion Cage

The Tool Box is an expansion card cage that permits use of the new Amiga 2000 cards on the A500 and A1000 computers. It provides 2 vertical 100 pin slots and will accept memory expansion cards, a A2088 IBM Bridgecard, hard drive controller cards including Commodore's A2090 and the Flash! Card, (as well as other A2000 specific cards). The Tool Box provides a bus return for future expansion, rear mounting panel, power status indicator light and one year limited warranty. Dimensions are 14 5/8D x 4 3/4W x 6 1/2H. The EEC-2520 and 2120 permits installation of a hard card type interface without additional power for the drive mechanism.

EEC-2510	A500 1 amp P.S.	**$189.95**
EEC-2520	A500 3 amp P.S.	**$259.95**
EEC-2110	A1000 No P.S.	**$179.95**
EEC-2120	A1000 3 amp P.S.	**$259.95**



ESCORT HD 32-48 MB *A500/1000*
DMA/SCSI Hard Drives

If your Amiga has a really voracious appetite, but your pocket book is on a diet, try our Escort 30Mb SCSI hard drive at a 20Mb price. Controllers, with a bus return are available for both the A500 as well as the A1000. Each accepts our 3.5 or 5.25 inch drive chassis (available in sizes up to 80Mb) with fan and power supply and is fully compatible with WorkBench 1.3. If you prefer buffet, buy the kit and supply your own drive. However you do it you'll enjoy lots of storage space at a great price.

Now you're cooking with gas!

Escort Hard Drive kits

A500/1000	32mb 3.5in 28ms	**$799.95**
A500/1000	48mb 3.5in 28ms	**$899.95**
A500/1000 **Kit (No Mechanism)**		**$335.95**

Sizes to 180mb - Call for prices

Feed your power hungry Amiga right, give it peripherals from Expansion Technologies.

** Toothpicks not included*

VISA and MasterCard accepted. Dealer Inquiries Invited.

Lunch for Lawyers; Amiga is a trademark of Commodore-Amiga, Inc. IBM is a trademark of International Business Machines Corporation. Escort is a trademark of Expansion Technologies Inc.

EXPANSION
TECHNOLOGIES

46127 Landing Pkwy, Fremont, CA 94538 415-656-2890

Amiga Genealogy?

Now Shipping Your Family Tree

THE FIRST GENEALOGY DATABASE FOR THE AMIGA

- Standard Reports • Free Form Memo Area
- Supports Over 10 Million Family Members
- Easy To Learn and Use

YOUR FAMILY TREE

Contact Your Local AMIGA Dealer or Send $49.95 + 6% Sales Tax (PA. Residents Only). $4.00 Shipping and Handling (Within U.S.A.) to:

MICROMASTER, INC.
1289 Brodhead Road, Monaca, PA 15061
PH. (412) 775-3000

AMIGA is a Reg. Trademark of COMMODORE AMIGA

In every case, he said, people had a positive attitude. "The exhibitors here were encouraged by the show, and they will support the future regional shows which are planned," Hunter said. "They're making suggestions, and we're listening to those suggestions. We have announced a show in Los Angeles the third weekend in May, and we expect most of the booths to be sold in the next couple of weeks."

Attendees also made their feelings known, in more ways than one. "People were spending more money here than at other shows, not only because they're charged less to get in, but because they get more for their money," Hunter said. "We plan to do surveys of the people who came to the show, to find out what they want. For instance, it would be foolish to have another show in Philadelphia if everybody's coming from New York," Hunter said.

Future plans include increased emphasis on seminars. According to Hunter, "This will require better planning further in advance; we're just now getting to know the exhibitors, It's like a first date." Attendees are encouraged to send their comments and suggestions to The Hunter Group, and anyone can receive a newsletter by writing to them at 204 Richmond Street West, Suite 410, Toronto, Ontario M5V 1V6.

A number of activities were available to show-goers, often simultaneously. Many demonstrations were held on the stage at one end of the show floor, and they utilized a huge, color CRT. Instructional meetings and press conferences were held in various rooms throughout the Civic Center. Support was also provided for user group meetings during the show.

When not being used for demonstrations, the large-screen monitor on the stage usually previewed eight new Commodore Amiga commercials to be aired on MTV and VH-1. These commercials are significant in that they were produced entirely on the Amiga and cost just $75,000. Commodore representatives estimated that such a group of commercials would normally cost more than one million dollars to produce.

Rundown

Here's a rundown of what we found to be new and interesting at the show:

Burocare Graphic Design, a company based in the U.K., showed AmiScan, a flatbed scanner with optical character recognition capabilities. The OCR software is capable of learning any font, and the system will sell for about $1600. Also shown was a software package that lets you connect the Polaroid Palette and Palette Plus (with RGBA inputs) to an Amiga 500 or 2000 with no intervening hardware. This package will be available at a substantially lower price than other systems now available. According to Burocare, "We couldn't get consistent results from the other package on the market, and we couldn't figure out why they needed hardware, so we decided to write our own software, and the results are now consistently better." Both items should be available by the time this article reaches print.

Moniterm, a company experienced in the IBM and Macintosh markets, unveiled the Viking 1 for the Amiga, a 19-inch grayscale monitor with 1008 x 800 resolution. The display is not interlaced, and the results are stunning. It's eerie to grab an Intuition window and drag it for what seems like several miles across the screen. Geared for high-end CAD and desktop publishing applications, the Viking 1 is driven by a card that uses the video slot in an Amiga 2000. There are currently no plans to support color, due to the way the Amiga blitter handles graphics. Moniterm says the main reason they decided to become involved in the Amiga market is that Commodore approached them asking for a large, high resolution monitor for high-end applications. Good work, Commodore! It would be nice to see this trend continue.

Comspec was showing their SCSI adapters which allow autobooting even on an A1000 running KS/WB 1.2. Recently upgraded for KS/WB 1.3 and

(continued)

TO WIN THE AMIGA HARD DISK CONTROLLER GAME... YOU'D BETTER HAVE A TRUMPCARD™

NOW SHIPPING!

THE *ACE* OF HARD DISK CONTROLLERS...

- SCSI INTERFACE SUPPORTS UP TO 7 DRIVES CHAINED TOGETHER
- AUTO BOOT ROM SOCKET IS READY TO GO WITH RELEASE OF 1.3
- SHORT A2000 CARD - TAKES UP ONLY 1/2 SLOT
- EXCLUSIVE IVS CHECKLIST SOFTWARE MAKES CONFIGURATION A BREEZE
- EXCLUSIVE IVS HYPERDISK PORT CONNECTS TO IVS *WARPSPEED* DMA CONTROLLER CARD TO SPEED DISK I/O UP TO 8X!
- SIMPLE, USER CUSTOMIZABLE DRIVER INCLUDES SUPPORT OF OVER 15 SCSI DRIVES
- COMPREHENSIVE DISK UTILITIES INCLUDING DISK PARTITIONING

LIST PRICE: $189.95!

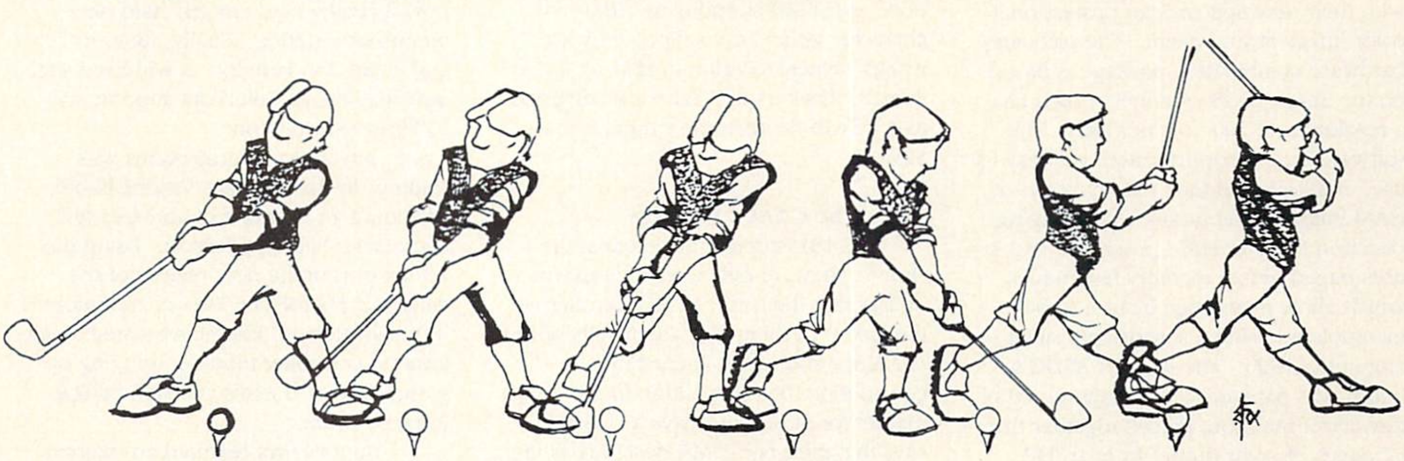
ASK YOUR DEALER TO CUT YOU IN ON THE BEST DEAL AROUND...

TRUMPCARD™

INTERACTIVE VIDEO SYSTEMS

15201 SANTA GERTRUDES AVE. STE Y102; LA MIRADA, CA. 90638 PHONE: (714) 994-4443

Don't Miss Your Best SHOT...



... read **Amazing Computing**™ for the most thorough coverage of the Commodore AMIGA. Amazing Computing is the leader in providing more solid information to the Amiga user. Isn't that why you read an AMIGA specific magazine? Read the original monthly resource, Amazing Computing, and see the difference a complete resource can make!

68000 DISASSEMBLY

OTG Software

ANNOUNCING . . .
DSM VERSION 1.0d

DSM is a full-featured disassembler for the Amiga. Check out these features and you'll see why programmers agree, "DSM is the best disassembler currently available for the Amiga, bar none."

- DSM disassembles any Amiga program that does not use overlays.
- DSM produces assembler-ready output that is 100% compatible with the Amiga assembler, assem. Excellent results can also be obtained using the C.A.P.E. assembler from Inovatronics.
- DSM features true text detection.
- Improved disassembly is supported using the built-in expert system mode.

If your local Amiga dealer doesn't carry DSM, send check or money order to:

OTG Software
200 West 7th Street Suite 618
Fort Worth, TX 76102

TX residents add 7.25% sales tax. Price $67.50

supplied with a menu-driven setup program, the several month old 2000 version was joined at the show by the new 500 model.

ASDG attracted much attention with their new SpectraScan professional color image input system. The software/hardware combination package is based on the Sharp JX-450 scanner, which has a resolution of 300 dots per inch. The software is very sophisticated, yet easy to use. It allows real-time manipulation of HAM images without any color fringing common to other HAM programs; and it uses paged virtual memory techniques which allow even huge images to be manipulated using a surprisingly small amount of RAM. The back of ASDG's booth was papered with a huge mural of the city of Madison, pieced together from 42 pages of color output from an HP PaintJet. The original picture—only 8 x 10 inches— was displayed alongside.

Spirit Technologies had the covers off all three Amiga models to show how their well-known memory expansions fit into the machines. Just beginning to ship at the time of the show, their ST506 interface allows cost-conscious Amiga owners to use an IBM style hard disk and controller. Although the Wedge and Palomax products have provided this capability for some time, the Spirit entry requires no "hardware hacking" and is nicely dressed up in an Amiga-beige box.

Also discussed but not shown was Spirit's MIDI Star, a new high-end MIDI interface with 2 INs, 6 OUTs, and an RS232 feedthrough. According to the representatives at the show, it should be shipping by the time you read this.

RGB Video Creations attracted the interest of video professionals with the CVC Video Workstation. This video post-production package consists of a special rack-mounted version of the Amiga 2000 developed by Computer Systems Associates, and a software package that can control up to 32 video decks at one time. This is a breakthrough system in that it has more features than dedicated professional workstations costing tens of thousands of dollars more. It fully utilizes the multitasking environment of the Amiga as well, allowing programs such as character generators to run concurrently. RGB Video Creations stresses support for their products—apparent from the tutorial module available for the workstation. The tutorial is similar to their Deluxe Help series of programs, and makes the system a breeze to use.

ATalk III, the newest telecommunications program from Oxxi, was debuting at $99.00, and is shipping now. ATalk III supports the IBM character set and emulation, allowing Amiga owners to call into IBM "color" boards. Owners of ATalk+ can upgrade to ATalk III for their old program plus $25.

The C Ltd. Challenge

C Ltd issued a challenge at the show: They are betting one thousand dollars that their new SCSI controller is the fastest of them all. The results of this challenge should be interesting, considering the competition in the Amiga market for SCSI these days. C Ltd also says that their non-DMA design is as fast as a DMA controller under most conditions, and faster at times. The new SCSI controller is scheduled to ship by January 1, and is slated to be the lowest priced SCSI controller on the market. In addition, the controller will allow multiple Amigas to be connected to a single hard drive, and it will allow multiple SCSI devices (Bernoulli boxes, optical disks, WORM drives, etc.) to be used.

NewTek had a large, bright display, but everyone missed Laura "M-M-Maxine" Longfellow. They announced Digi-View Gold, an upgrade to their popular digitizer, which requires no gender changer for the Amiga 500 and 2000. Two impressive demos played on their large monitor: an Allan Hastings animation and the unfinished "Demo Reel 2". Allan Hastings is now working with New Tek on a super secret hardware and software project that was developed while the Video Toaster was on hold. A New Tek style user interface will figure heavily into the new product. If the "Rush Hour" demo is any indication, the product will be drop-dead gorgeous. Any guesses?

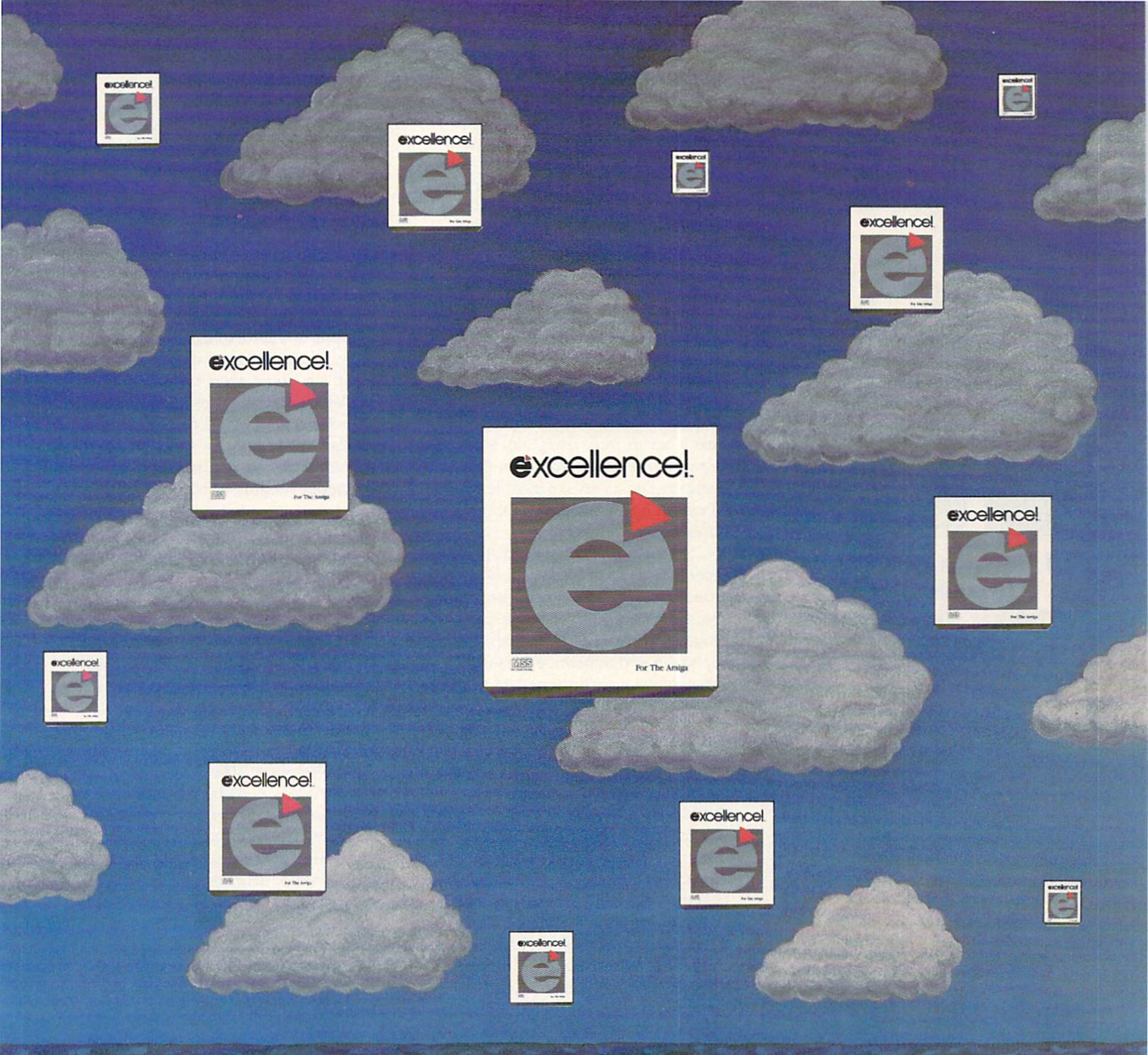
Toast in the New Year?

The Demo Reel, a showcase of New Tek's capabilities, combines three minutes of color video and stereo audio onto two disks. The demo was choppy at this stage, but what a compression ratio! The long-awaited Video Toaster should finally be released in February, priced in the $1200 to $1500 range. The design has been finalized, and the chips that have been a holdup are now available at a reasonable cost. It will be the first of a new product family, and New Tek is now freely releasing spec sheets. "It was ready to go before, but now its really ready to go," said one enthusiastic staffer. Finally, look for Digi-Paint 2 in January. It will have such new features as ColorFont support and 1000x1000 resolution.

Interactive MicroSystems was "hiding" in the Software Visions booth, providing an example of an extended MicroFiche Filer application. Using the ARexx port in the new release of the database, MicroFiche Filer communicated user selections to Interactive's MediaPhile infrared controller module, bringing up scenes from a laserdisc version of The Princess Bride.

Supra Corp. featured an internal 2400 baud modem for the Amiga 2000. It will be available in January for $179.95. They also plan to release an autobooting SCSI interface for the Amiga 500 at about the same time. Great Valley Products may have beaten them to the punch, however, as they were ready to

(continued)



THE SKY'S THE LIMIT

Don't limit your potential! Experience excellence!, a wordprocessor designed for your Amiga, with 250 available fonts, a Spell-As-You-Type 90,000+ word Dictionary, Grammatical/Style Checker, Thesaurus, Index and Table of Contents generator, Headers, Footers and Footnotes! Sail through PostScript output, True WYSIWYG, automatic Hyphenation, Math, beautiful resizable Color Graphics, flexible Mail Merge, Columns and an easy-to-use Macro-Language making complex actions a breeze! The fastest wordprocessor for your Amiga is the only one you'll ever need! Have an excellence! summer!

MSS
Micro-Systems Software

Committed to excellence since 1978

12798 Forest Hill Boulevard ● West Palm Beach, Florida 33414 ● 407-790-0770

See your local dealer or call our Sales Division
1-800-327-8724

Amiga is a registered trademark of Commodore Business Machines ● PostScript is a registered trademark of Adobe Systems, Inc.
execellence! is a registered trademark of Micro-Systems Software, Inc.

We take a byte out of the price
Not out of your pocket!

ONE
BYTE
(800) 441-BYTE

AMIGA
SOFTWARE
&
HARDWARE

51 Norwich New London
Turnpike Rte 32
Quaker Hill, CT 06375
(203) 443-4623

Amiga is a registered name of
Commodore Amiga, Inc.

ship autobooting SCSI controllers at the show. The GVP product is interesting in that the controller is built upon a 2-meg unpopulated RAM card. The card can be populated at any time by using 16 1-meg RAM chips, currently the most economical chip available.

Michtron demonstrated VIVA — the Visual Interactive Video Authoring system by Interactive Images — to interested showgoers. The demonstration consisted of a laserdisk system being driven by VIVA software, interfaced to a Commodore monitor with a touch screen. It was an excellent example of what the Amiga can do, given the proper environment.

Progressive Peripherals and Software was showing Ultra Design, their soon-to-be-released CAD program, on an Amiga 2000 equipped with a Flicker Fixer. This is the type of application that really shines on a flicker-free monitor. Trust us, if you've never seen a 640 x 400 non-interlaced display, don't even look at one unless you're ready to buy. The temptation is just too great.

Ultra Design is a very professional CAD program that not only supports every feature available in Amiga CAD programs today, but breaks new ground as well. Even with all this power, the program is simple to use. Each action you take is accompanied by unobtrusive context-sensitive help displayed in the title bar. An especially nice feature is the PasteUp system which allows complicated printing jobs, like murals, to be generated easily. This means you can now create a high resolution D-sized plot using a dot matrix or laser printer. The user interface got much attention, and it allows every task to be completed in clear, concise steps using either the mouse or the keyboard. Ultra Design is being created as an expandable product, so expect it to grow in several dimensions.

Precision Incorporated spread its American wings at the World of Commodore, debuting two new products and showing a third. The two new products available at the show were Superplan and Superbase Personal 2. Superplan is a sophisticated 74 function spreadsheet with macros, 18 different kinds of charts, and ARexx compatibility. ARexx compatibility allows the spreadsheet to communicate freely with other programs such as Superbase Professional. This integrated database/ spreadsheet capability is tough to beat.

Superbase Personal 2 is an upgrade to the original Superbase which benefits from intelligent data handling algorithms used in Superbase Professional. Although a "dongle" is still used for copy protection on Superbase Personal 2 (it has been dropped from Superbase Professional), an unprotected copy is available at very low cost when you send in your registration card.

Precision Audio

Precision also showed a product that marks a new direction for the company: Pro-Sound Designer. This is a high quality stereo audio digitizer with impressive features such as automatic gain control. In addition, there are many editing functions which advance the state of the art for audio digitizers on the Amiga. The product, priced at $159.95, should be available as you read this, and there will be an introductory offer allowing a fifty dollar trade-in on your current digitizer.

In the MindWare booth, representatives were debuting PageRender 3D, a new 3D rendering package scheduled for January release. Targeted at $159, PageRender 3D supports both Haitex and colored 3D glasses. Also being shown was PageFlipper Plus F/X, MindWare's upgrade to their PageFlipper animation package.

Brown-Wagh showed the latest upgrade to their Express Paint program. Although the program does not support HAM, just about every other feature is available. New features include a page size of 8000 horizontal pixels and vertical pixels (limited only by available RAM); sophisticated perspective mapping; color PostScript support; tinting, dithering, and smudging; extended gradient fill capabilities; and six ranges of color cycling. Stephen Vermeulen, the developer of the package, says that the program is aimed squarely at DeluxePaint. Some of the features of that program, he says, are awkward and hard to use. With Express Paint, many more features are available in an easier to use format.

ReadySoft showed two very exciting new products that will be ready for market soon. The first is the famous Dragon's Lair game by Don Bluth. It has been painstakingly brought to life on the Amiga with all the action and quality of the LaserDisk version. Eight hundred megabytes of data have been compressed onto six disks for this epic game. A different part of the game is featured on each disk, and the player works his way through a maze while each disk is loading, so that there is no break in the action.

AmigaMac?

In a totally different direction, ReadySoft also showed a hardware/ software product called the Mac Emulator to enthusiastic crowds. This technological wonder actually converts your Amiga into a fast Macintosh for about $200, including the price of the Macintosh ROMs. The hardware plugs into the external disk drive port and takes over the machine. (Pity... how many of us might dream of a 2000 with a bridgecard and Mac Emulator running a different computer's software in each of three windows!)

The Mac Emulator will run nearly every application available for the Mac, including Word, Excel, and Mac Draw. Ironically, Macintosh drawing programs run even faster on the Amiga than they do on an actual Macintosh, since the Amiga has the "blitter advantage." Now, anyone who has decided that they want a Macintosh (yes, Virginia, those people do exist!) can have one for less money, and they can get an Amiga for free! What a deal!

Gold Disk drew crowds with demonstrations of MovieSetter (their new animation package), ComicSetter, and Professional Draw (claimed to have 95% of the functionality of Adobe Illustrator). The debut showing of MovieSetter was complemented with demonstrations by James "Max" Morehead, one of the principle programmers. On several occasions, Max showed everyone just how easy it was to create an animation, putting together a complete scene with sound effects in a matter of minutes. (See "MovieSetter: A Chat with Max Morehead" for more information.)

All things considered—especially the fact that Comdex was scheduled for a mere two weeks later—the first U.S. World of Commodore show was an exciting four days and a definite success. We're already looking forward to the second show... and we hope to see you there!

•AC•

Templicity™

60 Ready-made Templates For Your Spreadsheet!

Available for Analyze!, Logistik, MaxiPlan, and VIP Professional

- •Easy to use
- •New! 1988 tax returns w/upgrades
- •Personal Financial Planning
- •Real Estate Decisions
- •Accounting
- •Credit and Loan Analysis
- •Personal and Business Budgets
- •Business Forms and Bookkeeping
- •Try for 30 days on our no-risk money-back guarantee.

Sample set-20 templates-$14.95
Complete pkg.-60 templates-$29.95
To order, send check/money order to:

The Sterling Connection
Box 4850
Berkeley, CA 94704
Specify which spreadsheet you have.

To order by phone:
Call (415) 655-2355
Mastercard and Visa accepted.
California res. add 7% sales tax.

Companies Mentioned

ASDG, Inc.
925 Stewart Street
Madison, WI 53713
(608) 273-6585

Brown-Wagh Publishing
16795 Lark Ave
Suite 210
Los Gatos, CA 050300
1(800) 451-0900

C Ltd.
723 E. Skinner
Wichita, KS 67211
(316) 267-3807

Comspec Communications
74 Wingold Ave.
Toronto, Ontario
Canada M6B 1P5
(416) 785-3553

Gold Disk
P.O. Box 789, Streetsville
Mississauga, Ontario
Canada L5M 2C2
1(800) 387-8192

Interactive MicroSystems
P.O Box 1446
HavenHill, MA 01831
(617) 372-0400

Michtron
576 S. Telegraph
Pontiac, MI 48053
(313) 334-5700

MindWare International
230 Bayview Drive
Suite 1
Barrie, Ontario
Canada L4N 4Y8
(705) 737-5998

Moniterm Corporation
5140 Green Circle Drive
Minnetonka, MN 55343
(612) 935-4151

NewTek, Inc.
115 W. Crane St.
Topeka, KS 66603
1(800) 843-8934

Oxxi Incorporated
P.O. Box 90309
Long Beach, CA 90809
(213) 427-1227

Precision Incorporated
8404 Sterling St.
Suite A
Irving, TX 75063
(214) 929-4888

Progressive Peripherals & Software
464 Kalamath St.
Denver, CO 80204
(303) 825-4144

Ready Soft
P.O. Box 1222
Lewiston, N.Y. 14092
(416) 731-4175

RGB Video
2574 PGA Blvd
Suite 104
Palm Beach Gardens, FL 33410
(305) 622-0138

Spirit Tecnology
220 West 2950 South
Salt Lake City, Utah 84115
1(800) 433-7572

Supra Corporation
1133 Commercial Way
Albany, OR 97321
(503) 967-9075

Roomers *& Replies*

by The Bandito

[The statements and projections presented in "Roomers" are rumors in the purest sense. The bits of information are gathered by a third party source from whispers inside the industry. At press time, they remain unconfirmed and are printed for entertainment value only. Accordingly, the staff and associates of Amazing Computing™ *cannot be held responsible for the reports made in this column.]*

There's more to some Amiga developers than meets the eye. The Bandito got a phone call in the dead of night from the informant known only as Deep C, who told a fantastic tale. It seems that early on in the Amiga's history, at least one major Amiga developer was financed by Commodore to create a software package that Commodore thought would be vital to their Amiga marketing efforts. This software package addressed an important, rapidly growing market segment . Commodore offered a sweetheart deal: they would pay for development, advertising, and marketing costs. The developer agreed, and as a result has had plenty of advertising paid by Commodore, special brochures, prominent position in Commodore ads, etc. In fact, Commodore pays for all the marketing and advertising of their software package. Not surprisingly, the software has come to dominate its market segment (although, to be fair, it *is* the best piece of such software).

This deal worked out so well for Commodore that they have approached several other companies to do software and hardware that Commodore would dictate the specs on. No takers so far, except guess who: the same developer they worked with before. Their new piece of software is designed to dominate an important market segment when it comes out in the near future. Why the secrecy from Commodore? Well, naturally, other developers might perceive that lots of dollars and prominent placement in Commodore marketing materials is unfair competition. (Amazing how such erroneous impressions can circulate.) Of course, part of that is sour grapes, because those developers weren't asked to be part of the deal (if they had been asked, the ones who complain the loudest now are probably the ones who would say yes the fastest). Until the software market for the Amiga really matures, Commodore will probably keep trying this strategy to fill in the gaps. Given the fact that Commodore hasn't yet succeeded in wooing the really big software companies to bring their products to the Amiga, this is the best thing Commodore can do for the future of the Amiga. But if any of the large software companies do enter the Amiga market, they might take a very dim view of this software subsidy. (A similar situation occurred with Apple Computer, which is why they spun off their software division as a separate company.) Come to think of it, that may be one reason why some of the software giants have stayed away from the Amiga so far. Something to ponder, eh?

HAM Paint Wars Update: A new version of DeluxePhotoLab is in the works, adding 3D perspective features among other things. No word on when it might appear, but the flowers will be blooming when it does, you can bet. Another interesting data point about the HAM Paint Wars: No advertising for any of the products lately, which could mean that the publishers are creating ads for new products. The Bandito is still waiting for the other contestants to come on down and join the fun. There probably won't be any new entries until early next year, which was about the time they were announced last year. You don't think that some of those companies were a mite premature in announcing their software, do you? Naaahhhh.

[*Informed sources at Electronics Arts indicated no work was being done on a version of Deluxe PhotoLab which supports 3D perspective.—Ed.*]

Speaking of premature announcements, the Video Toaster should win an award for the Most Anticipated but Delayed Product — it's been over a year now that NewTek first displayed it, and still no shipments. Information leaks picked up by electronic surveillance indicate that the Toaster is getting much closer to being reality, and is starting to be shown to a select few. In fact, a brand new Toaster tape was shown to the dealers at Comdex, and the picture quality looks much better than the previous showings, according to those who have seen it. The Bandito expects from all indications that the Toaster will appear before the groundhog sees his shadow in 1989. Reports are that it has many more features than the version previously shown, and the price tag will be around a thousand smackers. No word yet on exactly all that it does, but the Bandito's spies are busy putting psychoactive drugs into Topeka's water supply, so more data should become available.

[*Informed sources at NewTek indicate the Video Toaster will become available in March or April of 1989—Ed.*]

Comdex report: The usual Commodore trade show booth was in place, with the usual developers displaying the usual products. The most interesting news came from the seminars Commodore held for dealers. (The Bandito has made friends with some Commodore dealers. Normally, wildlife experts recommend that you keep well away from computer dealers and avoid feeding them, but the Bandito has discovered that if you catch them young and treat them right, they can make good companions.) The seminars were packed with more dealers than ever before, so it's obvious that Commodore is succeeding in expanding its dealer network. New 15 second TV spots created by Griffin-Bacall (Commodore's ad agency) were shown, but the reaction wasn't quite what Commodore had hoped for. These spots were created for MTV, and were far too hip for the mostly post-teen dealers. In fact, when a Commodore rep said "How about those commercials!" in an excited tone of voice, only about two dealers out of a thousand applauded. But even if the dealers were underwhelmed, the commercials will probably do well with the MTV crowd. It's also part of Commodore's strategy to reach out to the music dealers by getting to the clientele that goes to music stores.

The big Comdex news from Commodore was the announcement of the A2500 and the A2000HD. The A2500 is, as the Bandito may have mentioned before, an A2000 with a 68020 stuffed inside, 1.3 ROMs and 4 megabytes of RAM. Unfortunately, Commodore's price is very expensive (no word yet on what the street price will be). The A2500 should be available before Christmas, but you don't really need to wait for it—you can put one together yourself with an A2000 and third-party hardware, and a darn sight cheaper than Commodore's list price. The A2000HD is just an A2000 with a hard disk. The important thing about these announcements is that having "official" 68020 Amigas and hard disks helps the machine and the third-party hardware sell better. In effect, Commodore legitimizes these add-ons. Yeah, sure, it's cheaper with other people's components, but business buyers need the reassurance that "official" parts provide. So the dealers are very pleased that Commodore is listening to them and providing the machines the customers want — a good sign for the future of the Amiga.

It's clear from their statements at the show that Commodore finally sees the Amiga and the desktop video as hot topics. There are now (according to Commodore) over 1700 software packages available for the Amiga, and somewhere between 600,000 and 800,000 Amigas have been sold worldwide. The A2000 is now selling exceptionally well — it's even outdoing the A500 in some places. Commodore is clearly looking to the future where the Amiga replaces their C64 and PC clone product lines, and becomes a true "third force" to be reckoned with alongside of IBM and Apple.

Amiga's clunky external drives may be replaced by a slimline model next year, according to some stray bits collected from Westchester. This may or may not be the same as the higher capacity drive previously hinted at. Cynics believe it's just a new injection mold for the standard drive. The Bandito just wants a drive that doesn't click when it's empty.

Deep C reports that Commodore is planning to bundle a VCR with every Amiga 500 purchase for a limited time promotion for Christmas; the deala will also include some software packages. Another deal offers a MIDI keyboard instead of the VCR and a different set of software. Looks like Commodore is serious about moving hardware over Christmas, and thereby keeping those new dealers happy. It's a good deal for those of you about to buy an A500 anyway. Don't look for any price reductions after the promotion ends, because the Bandito hears that none are planned for the near future.

What's Uncle Jack up to these days? The word is that Uncle Jack no longer has an active role at Atari. He's semi-retired, taking it a little easier while his sons worry about the business. What the hey, he's rich enough that he doesn't have to worry. The Bandito should have such problems.

But Atari does have some troubles, what with Atari ST sales languishing in the U.S. While their videogame sales are strong, only a committed optimist believes that videogames will never slump again. So how does Atari plan to grow? Of course, there's been the rumors about a low cost UNIX workstation and the ABAQ transputer. But the Bandito thinks those machines will never appear, simply because that's not the type of market the Tramiels have gone after in the past. (Although word has reached the Bandito that Atari is looking for engineers with UNIX backgrounds, which would lend credence to that theory.) A new 4096 color Atari is in the works, and was shown to select groups at Comdex. But the price is substantially higher than an Amiga, and performance is not as good (no high-quality sound chip, for example). They'll have to work real hard to match Jay Miner's chip artistry.

A more likely occurence would be taking an Atari ST, stripping it down and making it into a videogame (as the Bandito reported that Commodore is considering for the Amiga 500). Remember, Atari has already done this with their 65XE (formerly an Atari 800), and it's done reasonably well. Of course, the market's getting crowded for 16 bit videogames — or is it? No one has released their machine in the U.S. yet. The first one to do so could gain extra market share simply by virtue of being first, which is something that the Tramiels wouldn't overlook.

On the other hand, maybe Atari will finally try to make some money with an IBM clone. Knowing Jack's Taiwan connections, he may well be the first to brave Apple's legal department and try to bring a Macintosh clone to this country.

Or how about an Amiga clone? Some Atari partisans would argue that an ST is already a Mac clone (with the addition of a Magic Sac cartridge), and there is an Amiga version of the Magic Sac on the way. But the Bandito has found out that even if the Amiga Sac comes out, it will have trouble with a good deal of Amiga software, particularly software that depends on the Amiga's hardware tricks (for instance, any program that supports HAM graphics). Doesn't sound like you'll be seeing an Amiga clone.

Falling RAM prices mean the return of reasonably-priced memory expansion, probably in time for you to spend the Christmas money you get from Grandma. Real techno-yuppies buy extra RAM just so they can run games from a RAM disk (now that's a status symbol). With RAM getting cheaper, Commodore is looking at the possibility

(continued)

Even Up The Score!

INVESTOR'S ADVANTAGE

Let your Amiga give you the Advantage in making *better investment decisions!*

Color graphics of Individual Stocks and General Market Trends help you make more profit in this volatile market. High Low Close, Moving Averages, Centered Moving Averages, Volume, Relative Strength, Stochastics, Wilder's RSI, Cycles, Trend lines and Momentum. Powerful reports such as the Relative Strength Report help you pick the best performers. Use the Market Barometers to help you time your market entries. Update Stocks, Mutual Funds and Commodities manually or automatically. Easy to use communications included.

Only **$99.95**

See your local Dealer or Call:
Software Advantage Consulting Corporation
37346 Charter Oaks Blvd
Mt. Clemens, MI 48043 (313) 463-4995

Amiga and the Investor's Advantage are trademarks of their respective companies.

of increasing the standard A2000 RAM to 2 or 3 megabytes, and bumping up the A500 to 1 megabyte. But don't hold your breath; you'll probably have to wait for an A2500 before more RAM is standard.

If you want AmigaDOS 1.4, be prepared to hang on until Spring 1989, and maybe later. Word is that they're adding more features, and debating exactly what should be included. Given the length of time it took to debug 1.3, we should see 1.4 before the Amiga centennial celebration.

The RAM chip crisis has been affecting some software companies. The supply of Nintendo cartridges is short because of RAM chips, and companies like Mediagenic, Broderbund, and Mindscape are not getting all the cartridges they want . All Nintendo cartridges are manufactured by Nintendo, so companies like Mediagenic must ship their software to Nintendo, order the cartridges from Nintendo, put up a lot of money in advance, then get a fifth of their order three months late. The recreational software market (for all computers) this year will total about $300 million dollars, while the market for Nintendo cartridges will be about $1 billion dollars. Doesn't take an Einstein to figure out which market developers would rather have their products in, does it? But Nintendo makes it hard for them by restricting the number of games they allow. That's part of the reason other companies want to jump in to the videogame hardware business: software companies will eagerly support them if it looks like they have a winner.

Nintendo's 16 bit videogame machine will be unveiled this December. Commodore will have private showings of its A500 "videogame" for certain special buyers, to gauge the reaction. What does all this mean to you? Probably more Amiga games, since "porting" to the Amiga will be easy and these new machines mean a big, juicy software market.

Speaking of the popular market, the rumor being floated by Commodore sales reps concerns the A500 appearing in Toys R Us. It appears to the Bandito that the salesdroids are testing the waters of dealer reaction to the idea. Of course, selling to mass market accounts was the thing that lost Commodore its previous dealer base back in the good old C64 days. This time, Commodore is hoping that the A2000 will be the machine for dealers, and the A500 will eventually be the mass market machine. It's possible this could begin sometime next year, but it's still a subject of heated debate.

Hot game titles in the stores these days: *Battlechess*, *Interceptor*, *Menace*, and *Rocket Ranger*. Look for some intense competition at Christmas time for your holiday dollar, with a lot of new releases and overseas imports. Amiga versions of the arcade hits *Star Wars* and *Empire Strikes Back* have appeared from a small company in England. They are faithful translations, and anyone who likes the arcade versions will want them. Be prepared to dig through the dealer's shelves to find the little CD boxes these games come in, but it's worth the hunt. This Christmas, if you like arcade games, you're in luck. But if you want to buy all of the arcade game titles for the Amiga, plan on getting a second mortgage on your house.

Despite all the new products, the big entertainment software companies are privately expecting a dull Christmas this year. The anticipated sales surge for October did not materialize, though this is perhaps due to the lackluster titles presented for the fall buying season. Some facts and figures about the software business: A good sell-in (that's the number of copies sold before shipment) of a new entertainment product is about 10,000 units for an Amiga title. Comparatively, a goo sell-in for a C64 or MS-DOS title is about 30,000. The biggest entertainment hit on the Amiga market has sold maybe 50,000 copies, while the biggest hit in the C64 or MS-DOS market has done about 500,000 copies. Perhaps this will make it clearer why it's important that there are more Amigas out there, and that there's less piracy. Even those publishers who love the Amiga have to make IBM versions to make the big money, and if Amiga versions can't make as good a return on investment, they may stop doing Amiga products. So convince all your friends to get an Amiga and buy software, since that will mean more and better software for everyone.

(continued)

TxEd *Plus*™

The Text Editor for the Amiga™, *Plus* a whole lot more.

BlitzDisk	Disk cache, speeds up floppy and hard disk reads up to 2000%.
FastFonts	Speeds up text display.
FunKeys	Hotkey window manipulator.
ARP	Latest versions of the AmigaDOS™ Replacement Programs.
AREXX	Demo version of the AREXX, the macro processor used by TxEd Plus.

TxEd *Plus* is a powerful yet simple text editing environment designed to meet all of your text editing needs. It's small, fast, and fully configurable. The AREXX macro language connection makes TxEd *Plus* an ideal foundation for your complete multitasking system.

"As a programming and general-purpose text editor, TxEd Plus sets the standard."
-Warren Block, Sept '88 INFO Magazine

"(TxEd V1.3 is) ...a very good editor and an excellent value."
-Jan&Cliff Kent, Vol. 1 #9 Amazing Computing

"FastFonts / BlitzDisk provides much more than Facc II for a similar price."
-Warren Block, July '88 INFO Magazine

"Yes, get TxEd if you're editing almost anything on the Amiga."
-Bruce Webster, July '86 BYTE Magazine

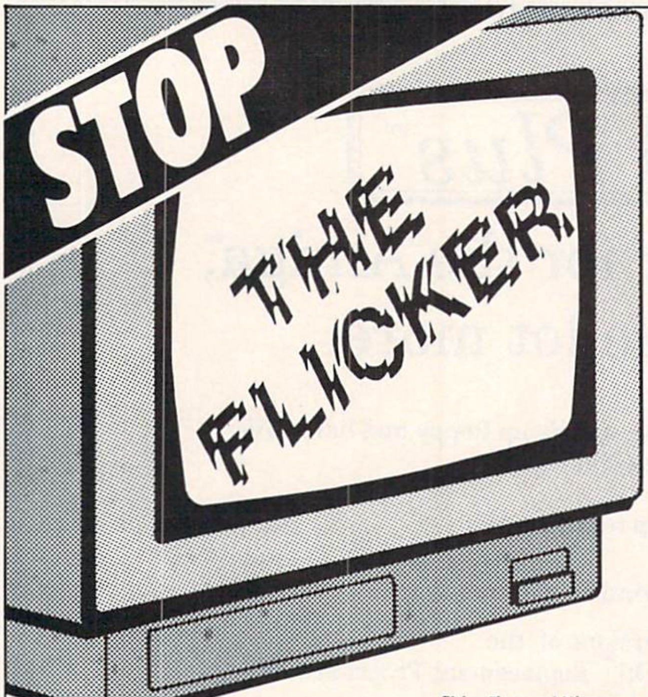
"I found the ARP commands to be smaller, faster, and more powerful than their BCPL forerunners."
-Jeff Blume, July '88 AmigaWorld

TxEd

$79.95

Microsmiths, Inc
PO Box 561, Cambridge MA 02140
(617) 354-1224 BIX: cheath CIS: 76004,1766

MC and Visa accepted. Mass Residents add 5% sales tax.
Amiga and AmigaDOS are trademarks of Commodore-Amiga, Inc



flickerFixer eliminates your Amiga 2000's interlace flicker and visible scan lines. The result: **superior quality color or monochrome graphics and text** — for a full range of demanding applications, including CAD, desktop presentation, graphics, animation, and video.

Reviews are impressive: Commodore Magazine 12/88: **Best of 1988 Award.** AMIGAWORLD 12/88: **#1 Readers' Choice Award.** Amiga GURU 5/88: *"The display is fantastic . . . It is the best display we have ever seen on any computer system."*

flickerFixer fits into the Amiga video slot, is fully compatible with all user software, and does not modify the standard Amiga video signals. The board upgrades the Amiga 2000 with a flicker free 4096 color palette, has an overscan mode that features a screen size of 704 x 470 pixels and drives most of the popular PC Multiscan and VGA monitors, including the NEC Multisync and Mitsubishi XC1429C.

*flicker***Fixer**™

Advanced Graphics Adapter For The AMIGA® 2000

flickerFixer is priced at $595. It is made in the USA and is FCC Class B approved. For more information or to order, call MicroWay Sales at (508) 746-7341 or your Amiga Dealer.

flickerFixer and **Microway** are trademarks of Microway, Inc.
Amiga is a registered trademark of Commodore.
Multisync is a registered trademark of NEC.

Microway

P.O. Box 79
Kingston, Mass. 02364 USA
(617) 746-7341

32 High St.,
Kingston-Upon-Thames, U.K.,
01-541-5466

58 Atchison St.,
St. Leonards, NSW, Australia
02-439-8400

Piracy is concerning more and more Amiga developers these days. Some developers estimate that they're losing 50% of their sales to pirates. (The Bandito thinks it's interesting that the publishers of a major copy program for the Amiga have dropped their support for the product now that they produce games.) Publishers would rather not copy-protect the disks because of the growing number of hard disk owners out there, but the lost sales may force them to do so. Worse yet, it could force some publishers out of the Amiga market entirely.

If you don't think that can happen, take a look at the Atari ST market, where piracy grew so bad that the software market collapsed (consequently, so has the hardware market). Some developers have tried putting in a code wheel or a map with a game, but that doesn't help enough. The Bandito hears that things may get serious soon, with some of the major, well-known pirates (you know who you are) being the object of legal action. Developers would like to see the pirates behind bars, and that just may happen in a few cases.

Speaking of software, DeluxePrint II has shipped — is the Bandito the only one who noticed?

Gold Disk's Professional Draw package looks like another important product for the Amiga — a full featured draw program that uses Postscript for high-quality output. Those who've seen it say that it beats the bits out of Adobe Illustrator for the Macintosh, and ProDraw offers color support, besides. Yet another good reason to buy an Amiga instead of a Macintosh.

In fact, if you stack up a Macintosh II against an Amiga 2000 with similar equipment, the Amiga competes extremely well; when you throw in the price, it's no contest. One difference that you can't yet overcome is the number of colors available; the Macintosh boasts a palette of 16 million colors, and with some expensive boards ($2500) you can even use all of them (or at least as many as you have pixels on your screen). The Bandito hears that some Amiga developers are aiming to fix this lack on the Amiga's part. The answer may be expensive, but you will be able to get millions of colors on your Amiga, and perhaps even larger screen sizes. Look for these solutions to arrive next spring, but expect to pay more than $1000 for the privilege.

While the Amiga developers are working on getting a larger palette, Apple knows that Amiga's desktop video power is a major selling point. The Macintosh II is a poor machine for video because the signal it puts out is far away from NTSC, and must undergo considerable (and expensive) manipulation to become NTSC compatible. The Bandito has learned through his sources that Apple has created a new generation video chip that will make future Mac II's much more video capable (though still not as compatible as an Amiga). Of course, they'll still be much more expensive than an Amiga, but it's clear that Apple intends to narrow the technological gap to compete more strongly in the growing desktop video market.

•AC•

[*In an attempt to verify the above comments concerning Commodore, we have notified them of these comments via FAX. As of December 23, 1988, we have not yet received a reply. Commodore's comments will be published in Roomers when a reply is received.—Ed.*]

PD Serendipity

Insight into the World of Freely Redistributable Software for the Amiga™

We're back, with another exciting installment of PD Serendipity! This month we'll cover Fred Fish 163-172. So let's go!!

Fred Fish Disk 163

Bankn V1.5
Update to FF 120
A complete checkbook system offered by the author as shareware. By Hal Carter

FiveInLine
A board-playing game similiar to Go-Moku, Ristinolla, etc. Fast-paced and quickly addictive! Includes the source.

MachII V2.4c
Update to FF 130
A "mouse accelerator" program that also includes hotkeys, the features of sun mouse, clicktofront, popcli, title bar clock with a bbs online charge accumulator, and more. Includes the binary only. By Brian Moats

MemTrace
Routines to help debug memory allocation and freeing during program development. Will complain if you try to free memory you didn't allocate and will report on memory not freed when your program finishes. By Jojo Wesener

PcPatch
Patches for PCCopy and PCFormat from the 'EXTRAS 1.2' disk, to allow reading/writing/formatting 3.5 inch 360k (2 sides/40 Tracks/9 sectors) MS-DOS disks. By Werner Guenther

ReadmeMaster
A nifty little database for finding those programs that you know exist somewhere in the AmigaLibDisk library. Maintains a key-word dictionary of the Contents descriptions that allows searching by disk number, program title, author's name, or some other descriptive word. Currently supports disks 1-154 with planned updates from the author. Binary only. By Harold Morash

View
A mouse-oriented text file reader. Sample operation is demonstrated in reading the View.doc file, instead of using the usual "Less" text file reader. By Bryan Ford

Fred Fish Disk 164

C-Functions
A group of four little C-functions to add to your library to make your programming life a little easier. Includes source and a small demo program showing some of the results. By Lars Thuring

DiskSalv V1.3
Update to FF 20
Very useful program to recover files from a trashed AmigaDOS disk. Can also "undelete" files deleted by mistake, so long as they have not overwritten by further disk activity. Requires two disk drives. Binary only. By Dave Haynie

Hed
A handy little editor that is more user-friendly than "Ed", yet doesn't require the memorization of complicated keystrokes of some of the larger, more powerful, editors. Binary only. By Hal Carter

Newton V1.0
Uses the "Newtons Method" algorithm to estimate both real and imaginary roots of a polynomial of degree 20 or less. includes source. By Daniel Barrett

NewZAP V3.18
Update to FF 58
A third-generation multi-purpose file sector editing utility, from the author of FileZAP. Displays and edits full 512-byte sectors via a 106 character wide internal font. Includes a search feature to find specific strings or hex digits, forwards or backwards. Binary only. By John Hodgson

PcView
Provides the PC community with the opportunity to display IFF pictures to the best of EGA's ability. Displays Amiga pics, IBM-PC Deluxe Paint Pics, Apple II-GS Deluxe Paint Pics, and others in the IFF standard format. Includes the source. By John Hodgson

PolyRoot V2.00
Another Polynomial root-finder using the Newtonian algorithm. Nicely done in AmigaBasic with good documentation file. By Jon Giorgini

PrtDrivers
A couple of new Printer Drivers. One for Digital Equipment's LN03+ laser printer, and one for Mannesmann Tally's MT420d dot matrix. Authors: DEC LN03 — Bernie Mentink MT420d — Sascha Wildner

(continued)

Wilkommen

Bienvenue

Welcome

All Foreign Computer Stores / Magazine Dealers

Welcome to the amazing world of Amazing Computing™ - one of America's leading magazines for Commodore Amiga users. Amazing Computing™ was the first magazine to document CLI, the first magazine with a 1 Meg Amiga upgrade hardware project, and the first magazine to offer serious programming assistance.

Don't deny your large English-speaking audience the kind of magazine that will provide them with complete information on the Amiga. Come and join our Amazing world as either a dealer or an advertiser.

For details on becoming an Amazing Dealer or an Amazing Advertiser, please contact:

Marie A. Raymond
International Coordinator
PiM Publications
P.O. Box 869
Fall River, MA 02722
(508) 678-4200
(800) 345-3360
FAX: (508) 675-6002

Amazing COMPUTING

Your Original AMIGA Monthly Resource

Zoo V2.00

Update to FF 136

A file archiver, much like "arc" in concept, but different in implementation and user interface details. Includes some nice features that "arc" lacks (such as file/path names up to 255 characters in length). Binary only. By Rahul Dhesi, Amiga port by Brian Waters

Fred Fish Disk 165

Conman V1.3

Update to FF 133

Extremely useful replacement for the standard console handler, provides line editing and command line histories. Completely transparent to any application program that uses CON: windows. This program is shareware, and well worth a donation to the author. Binary only. By William Hawes

CPM

Another CP/M emulator independently authored from the version that appeared on disk #157. Emulates a CPM computer with a Z80 processor connected to an ADM3A terminal. Assembly source included. By Ulf Nordquist

Parsnag

A program to aid in performing color separations on Epson JX-80 printers. Includes source. By John Hodgson

PlotView

A couple of programs, Plotview and Plot2Am, for viewing UNIX plot files. Also included are two sub-directories: Plot — a device independent plotting package for the Amiga, compatible with the UNIX plot subroutine package and Plot2Tek — converts UNIX plot format files to Tektronix 410x terminal graphic commands. Source included. By Joel Swank

RamCopy

A copy program designed for machines with 1 meg or more of Ram and only one disk drive. Allows you to copy a complete disk in only one pass. By Stephen Gunn

SPUDclock V1.2

A simple program that uses the narrator device to speak the time at certain user specified intervals. Lots of command line options. includes source. By Robert E. Beaty and H. Bret Young

Fred Fish Disk 166

AutoGraf V1.0

Collects and graphically displays information on auto mileage. Features such as miles per gallon, cost per mile, miles driven, highs, lows, averages, etc. Includes sample data file, a couple utility programs and source. By Joel Swank

Cref

Update to FF 103

A C cross referencer program. Prints out your code with line-numbers and complete key-word cross-referencing. Includes source. By Mike Edmonds; Amiga port by Joel Swank

MultiCalc

Yet another RPN type graphic calculator. This one generates answers with extreme precision (if 3000 digits is enough!) Features a 48-digit scrollable display, mouse driven with lots of keyboard shortcuts, and iconification during non-use. Binary only. By Kenneth Johnson

Stevie V3.10a

A public domain clone of the UNIX 'vi' editor. Supports window-sizing, arrow keys, and the help key. Version 3.10a, includes source. Amiga port by Tony Andrews

Fred Fish Disk 167

CDecl

Update from FF114

English to C (and vice versa) translator for C declarations. This little gem will translate english such as "declare foo as pointer to function returning pointer to array 10 of pointer to long" into "long *(*(*foo)())[10]", and vice versa. Includes source. By Graham Ross with enhancements by David Wolverton, Tony Hansen, Merlyn LeRoy, Udi Finkelstein and probably others!

CLIcon

Allows you to run cli programs from the workbench, similar in operation to IconExec, but more versatile. By Bryan Ford

NAG® PLUS 3.0 **Schedule Assistant**

Software for the AMIGA™ computer

Voice and Sound Reminders - *Create your own or use NAG PLUS library.*

Perpetual Calendar *-Enter 99 events per day.*

Auto Dialer *-Connects YOU, or your modem.*

AREXX Port *-Commands any timed event or action.*

Notepad *-Click on any word to open text editor.*

Print *-One click prints any file or appointment list.*

Suggested Retail $79.95. Ask your dealer or contact:

Gramma software For Your Own Good

17730 15th Avenue N.E.
Suite 223
Seattle, Washington 98155
(206) 363-6417

CloseMe
Another ingenious perversion in the screen hack category. Don't miss this one...surely destined to become a classic! Includes source. By Charlie Gibbs

DSM
(Dynamic Sound Machine) Demo version with "Save" disabled of a program that will take any IFF sound or raw data and save it as a totally self-contained, runable program. By Foster Hall

MRPrint V3.1
A cli-based text file printing utility with lots of nice features, including tab-to-space expansion, page headers, line numbers, margin control with line-splitting and pagination correction, ARP wildcard support, and auto- rejection of files containing binary characters. Includes source. By Mark Rinfret

Smus3.6a
Update to FF 58
An enhanced version of the smus player that last appeared on FF58. By John Hodgson

Sounddemos
Some very nice demos for showing off the incredible audio power of the Amiga!. 100% assembly language, make sure you have the stereo connected for these! By Foster Hall

Fred Fish Disk 168

This is the also the first disk of a two disk "Matt Dillion special" containing binaries and sources to the latest versions of a great deal of Matt's many programs.

Fred Fish Disk 169

This is the also the second disk of a two disk "Matt Dillion special" containing binaries and sources to the latest versions of a great deal of Matt's many programs.

Because of the nature of Matt's programs and the need for his support libraries to recreate many of his programs, Fred opted to keep these disks pretty much as Matt sent them to him, rather than follow the usual AmigaLibDisk format of binaries and sources in one directory. This would have undoubtedly increased this to a three disk set. P*lease note that if you plan on recreating any of these programs, then it is quite possible that you will need information from both disks.*

1988 TAXES

Calculate your 1988 Taxes with Quality Business Systems' TAX PLAN
FAST, EASY, ACCURATE

- Prints Forms
- Audits Taxes
- Multiple Returns
- 20 Forms and Schedules
- Saves to Disk
- Forecast 1989 Taxes

Commodore-Amiga
ONLY $34.95
Add $4 S/H, MA add 5% Sales Tax. Check, M.O., Credit Cards

Quality Business Systems
P.O. Box 805, Hudson, MA 01749
1-800-225-5800

Fred Fish Disk 170

Aftterm
Commmunications program utilizing IBM 3278 terminal emulation. Binary only. By Don Brereton

Dis6502
A ported 6502 disassembler with support added for C64 binary files. Includes source. By Robert Bond, Amiga port by Udi Finkelstein

FastText
Blitter based fast text rendering routines written in assembly. Unique in the fact that they speed up rendering of non-proportional fonts of any height, and from 4-16 pixels in width. Source and test program included. By Darren M. Greenwald

MRBackUp V2.4
Update to FF129
A hard disk backup utility that does a file by file copy to standard AmigaDOS floppy disks. Includes an intuition interface and file compression. Binary only. By Mark Rinfret

PtrAnim
Nifty pointer animation program, includes lots of samples, a utility program and instructions on creating your own animations. Be sure to also read the "Disclaimer" in the author's ReadMeFirst file. Binary only, shareware. By Tim Kemp

Surf
Generates bezier surfaces of revolution. Will produce some amazing pictures of wineglasses, doorknobs, or other objects one could turn on a lathe. Includes the capacity to map IFF image files onto any surface that it can draw. Source included By Eric Davies

(continued)

MOVING?

SUBSCRIPTION PROBLEMS?

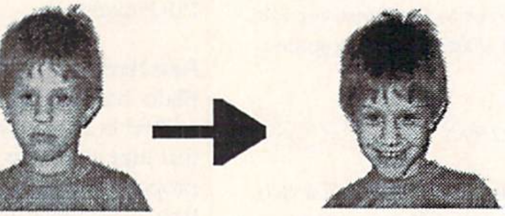
Please don't forget to let us know. If you are having a problem with your subscription or if you are planning to move, please write to:

Amazing Computing Subscription Questions
PiM Publications, Inc.
P.O. Box 869
Fall River, MA 02722

Please remember we cannot mail your magazine to you if we do not know where you are.

Please allow four to six weeks for processing.

EDUSOFT™



Enlighten Your Child

Introducing........

Crayons And Rainbows

The Amazing Talking Computer teaches your preschooler about colors as your child plays a game to help the crayons paint a rainbow. Animated characters offer many challenges in color recognition and identification by providing entertaining scenes and situations.

Order our preschooler software from us or from a dealer near you.

The Trumor Company
299 Coon Rapids Blvd NW
Suite 103
Coon Rapids, MN 55433
(612)785-2289
An all Amiga company

For All Amigas. 512K memory required. More recommended.
AMIGA is a trademark of Commodore - AMIGA, Inc.

Turbo

Opens a small window with a gadget that when selected, turns off bitplane, sprite, copper and audio DMA, presumably to increase system speed. Includes source. By Oliver Wagner

Fred Fish Disk 171

AZComm V1.00

Modified version of Comm 1.34 that contains Zmodem send, receive, and resume receive. Binary only. By SS. Patel

Maze

A couple of very nice demos for the creation and use of single-solution mazes, one of which is practically a stand-alone game. Includes source. By Werner Gunther

Sozobon-C

Atari ST version of what appears to be a full K&R freeware C-compiler, assembler and linker. The compiler main pass and the assembler were compiled and tested on an Amiga A2000 with only minimal changes, and they appear to work (to the extent that they believe they are running on an Atari-ST), so an Amiga port should be relatively easy. By Sozobon, Limited.

Xoper

Very comprehensive program to monitor and control system activity. Monitor cpu, memory usage, ports, interrupts, devices. Close windows, screens, show loaded fonts or last Guru code number. Clean up memory, flush unused libraries, devices, fonts. etc. and a whole bunch more! Spawns its own process. A very handy background task to have loaded. Assembly source included. By Werner Gunther

Fred Fish Disk 172

DataToObj

A utility to convert raw data files (sprites, image data, text, etc.) directly into object code which can then be linked to the main program without the need to go through the compiling process. Includes source. By Werner Gunther

Handshake V2.12a

Update to FF60

A full featured VT52/VT100/VT102/ VT220 terminal emulator. he author has taken great pains to support the full VT102 spec. Now supports ANSI colors, screen capture and more. Binary only, shareware. By Eric Haberfellner

MFix V1.0

A small program to insert in the startup-sequence of the commercial program, Marauder II, from Discovery Software, International. Whenever the copy process is started, the rainbow screen is covered by a bare screen until the copy is finished. The author claims a 25% decrease in copy time is achieved. Binary only. By Stephen Gunn

PopInfo V2.0

A "shrinkable" workbench utility to show you some info that Workbench doesn't, such as free memory on external devices, chip, fast, and total ram usage and more. Includes source. By Jonathan Potter

ProCalc V1.2

Update FF139

A program that simulates an HP-11C programmable calculator. Lots of enhancements and bug fixes since the original version on FF139. Binary only, shareware. By Gotz Muller

Spiff

Make controlled approximations between two files. Similar to "diff" but more versatile. Allows for the handling of numerals as string literals or numeric values with adjustable tolerances. Provides for embedded commands, scriptfiles, and many other command-line parameters. Potentially very useful, but needs some Amiga-specific work. Source and some example files included. By Dan Nachbar, Bell Communications Research (BELLCORE)

•AC•

New Cli Window

1>

THE COMMAND LINE

AmigaDOS Replacement Project (ARP)

by Rich Falconburg

AmigaDOS commands allow a lot of control over the Amiga's various functions and operations. Unfortunately, the inconsistent templates for those commands can cause some frustration. Because of a curse called BCPL (a programming language used to write much of AmigaDOS) programmers have to re-invent the wheel to accomplish seemingly simple tasks. Charlie Heath and his band of magicians at MicroSmiths Inc., took on the monumental task of correcting some of these deficiencies. The result is the AmigaDOS Replacement Project. They have standardized the command templates and where possible, improved or added multiple argument and wildcard support and provided abbreviations to command switches.

Those lamenting the loss of the asterisk (*) as the universal wildcard take heart. ARP supports it, in addition to the normal AmigaDOS format. Because the replacement commands are nearly identical to their AmigaDOS counterparts, it won't be necessary to start from scratch and learn a new environment. The compatibility is such that if you don't use any of the new features provided by the ARP commands (inconceivable!) you probably won't notice any change.

Something you may notice is the difference in the size of commands. It's fascinating to copy more commands to a Workbench disk and watch the free space on that disk increase! All of the new commands are written in Assembly Language which makes them significantly smaller. The collection will include a new library that is required by the commands and must be copied to the LIBS: directory. This file, arp.library, is also partly responsible for the smaller size of the commands. When you copy the ARP commands to your Workbench disk you will have an extra 20k Bytes of free space. New and Improved!

Several new commands added by ARP but before we get to them let's look at some of the improvements made to the old ones.

ADDBUFFERS
You can now add new buffers to DF0: through DF4: with a single command line.

ASSIGN
As well as being able to make several assignments at once you can remove an assignment by using a dash as the logical name.

CD
Allows wildcards to be used and will set CD to the first qualifying match. A new switch, ROOT, will force a CD to Drive 0, not the Volume in the drive.

CHANGETASKPRI
Now allows you to select the Task number to change.

COPY
To define the current directory as the destination use the asterisk (*) (AmigaDOS allowed “”). A new switch, FLAGS, controls the copying of FileNotes, the date, and the protection bits with the file. (See the discussion on Environment Variables for more information.)

DATE
The addition of an environment variable allows changing the display format to one of four varieties. See the section on Environment Variables for more information.

DELETE
You may specify any number of files or directories and use wildcards in any position. A new switch, ASK, provides a confirmation request allowing you to answer Y or N to delete each file displayed.

DIR
Wildcards and filenames may be used and options have been expanded. These include OPT F - files only, OPT H - print directory names in reverse video, and OPT S - print the sizes in bytes.

ECHO
No longer requires quotes and will recognize escape sequences in both quoted and unquoted strings.

INFO
This command now displays the actual formatted capacity of the volumes shown. Devices are now shown in ascending order regardless of the order they were mounted in.

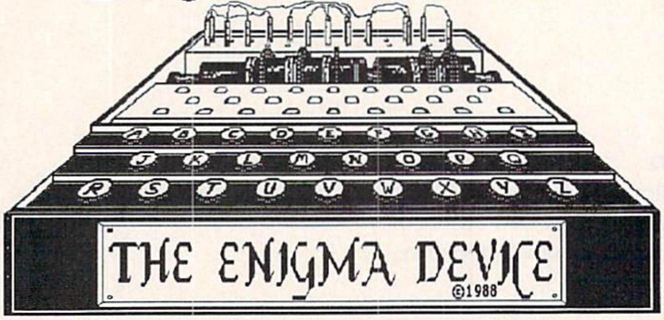
JOIN
You may use wildcards and select any number of files to be joined together. Effectively obsoleted by ARP's TYPE command.

(continued)

King Publishing

PRESENTS

Gaylan Wallis's



A GAME OF CRYPTOGRAMS
For Amiga™
NTSC/PAL SRP $39.95

- Over 1000 Puzzles Included
- Create Your Own From Any Source
- Automatic Tracking Of Player Progress
- Supports Up To -20- Players With Histories
- Beginner & Advanced Help With Cheat Button

Phone 712-252-4604
King Publishing 921-32nd St. Sioux City IA 51104

LIST

Wildcard pattern matching of the filename/directory is now supported, making the PAT and S switches unnecessary even though they are still included. You may now get a sorted listing with the new switch SORT. The date format displayed is determined by an environment variable.

MAKEDIR

Create multiple directories with one command line.

MOUNT

This command now allows more than one device to be specified. A new switch, STARTPROC, will also load the device handler at the same time. The handler is normally loaded on the first access of the device.

PATH

Now allows as many definitions as you wish.

PROMPT

Will recognize two special characters - %N, provides the current process numbers and %p which works with the ARP CD command to display the current directory. If the prompt string exceeds 69 characters, this value is ignored until the length is less than 70 characters.

PROTECT

This command now supports wildcards. If wildcards are used, the name of the file and the protection level it's being set to will be displayed. If the QUIET switch is added, then that information will not be displayed. Unlike AmigaDOS, many of the ARP commands will check the protection bits and respond accordingly.

RENAME

Now supports wildcards in various combinations allowing you to move several files at a time. The wildcards also make it easy to substitute various parts of the name while preserving or adding to the rest. This command is a BIG improvement over its AmigaDOS counterpart.

SEARCH

Now supports wildcard specification in the search string.

SETDATE

Supports wildcards and multiple arguments and the date format environment variable. If no date is specified the current date and time is assumed.

SORT

This command now provides the CASE keyword to allow you to make the sort case sensitive. You can also specify the number of characters to check per line. No more crashes with this one. If there is not enough memory to do the sort, it tells you so.

TYPE

Type supports multiple filenames and wildcard specifications. If a wildcard is used, the name of each file is shown as it is displayed. TYPE entered without a file argument— TYPE TO df1:Tempfile—will cause it to send keyboard input to the TO destination. To terminate the send or close the file enter Control \.

VERSION

Displays the current Kickstart, Workbench, and ARP version numbers.

New Commands

Normally, when you run a program, it is loaded into memory each time it is called. If several processes use the same program, a new copy of that program is placed in memory for each process. This is most common with commands executed from the CLI. The RESIDENT command works in concert with the ARUN command to help reduce the memory overhead used by programs. RESIDENT allows only one copy of the program in memory at a time regardless of the number of processes using the program's code. To make some of the most common commands Resident you would enter:

```
1> RESIDENT C:DIR C:COPY C:RENAME C:TYPE C:DELETE
```

RESIDENT stores only the last component of a path specification. If you want more information about the commands that have been made Resident use the FULL parameter.

```
1> RESIDENT FULL
Name            UseCount CheckSum Segment
DIR                    0 72E67357 00310C35
COPY                   0 8F2D2AB4 00312E1B
RENAME                 0 BE5D4C50 00313041
TYPE                   0 3F6BEB53 0031312B
DELETE                 0 FD8A62EC 0031324B
```

To remove a program from the Resident list use:

```
1> RESIDENT DEL DIR
```

If you use wildcards, each program name will be displayed as it is added. To suppress this, use the QUIET switch. Some programs cannot be made Resident. If so, RESIDENT will make the determination before running the program and post a requestor to inform you there is a problem.

The ARUN command is an enhancement to the normal Run command and allows you to set the size of the stack, the priority, and detach status for the new process. A detached process will not force the parent process to remain open until it is finished. To start a detached program with a priority of -5 and stack size of 5000, use:

```
1> ARUN DEMO NOIO STACKSIZE 5000 PRI -5
```

Although ARUN does much of the same work that the AmigaDOS RUN command does, it cannot as yet completely replace it. Because ARUN does some things a bit differently, it may be incompatible with certain programs that rely on features unique to RUN. Actually it's more of a programming problem than a usage problem and is related to some of the internal functions of AmigaDOS. You may have seen some documentation notes with various programs that urge to insure that Run is in the C: directory. This indicates a program that would have problems with ARUN.

Special features of ARP commands allow ARUN to change the stack size for them automatically. This helps prevent system crashes and makes memory usage more efficient.

A new command that is mostly useful to programmers is the LOADLIB command. However, with all the new run time libraries showing up, this command might come in handy. For example, if you decided you would not need one of the libraries on your boot disk because you rarely use the program that needs it, you can load it in when needed from the LIBS/ directory of another disk with LOADLIB. Because LOADLIB cannot use the arp.library, it does not support some of the standard ARP features such as the multiple argument feature.

Environment Variables

Okay, so ARP commands use environment variables. What's the big deal? Flexibility and ease of use. The SET command is a spinoff of the UNIX™operating system. Many of the Amiga developers have experience with that environment so it's no surprise to see the Amiga environment grow more similar to UNIX every day. (Stop groaning. Where do you think most

(continued)

BREAKTHROUGH

All Hard Drive Back-Up Programs were created equal . . .

UNTIL NOW!!

EZ-Backup

100% better than any other hard drive back-up program

EZ-Backup is a genuine breakthrough

EZ-Backup actually manages the space on your backup disks--

Before EZ-Backup the only way to get rid of the piles of incremental back-up disks that littered your desk was to do another full back up. This method is inconvenient and requires a whole new set of unformatted floppy disks.

With EZ-Backup you use the same set of disks for every incremental back-up. Only one full back-up required--Ever! Space on the disks is managed by deleting obsolete archive files and allowing you to save from 0-255 versions of each file. **Your files are saved--even if you have completely deleted them from the hard drive!**

EZ-Backup comes with an optional warning screen--
We all tend to put off doing backups. EZ-Backup's warning screen reminds you. If you would rather not be reminded--you have the option to shut the warning screen off.

EZ-Backup prevents you from damaging valuable data--
By checking the volume label, EZ-Backup keeps you from writing over important files.

EZ-Backup uses Standard Amiga format--
Files are archived in standard Amiga format and work with all standard utilities.

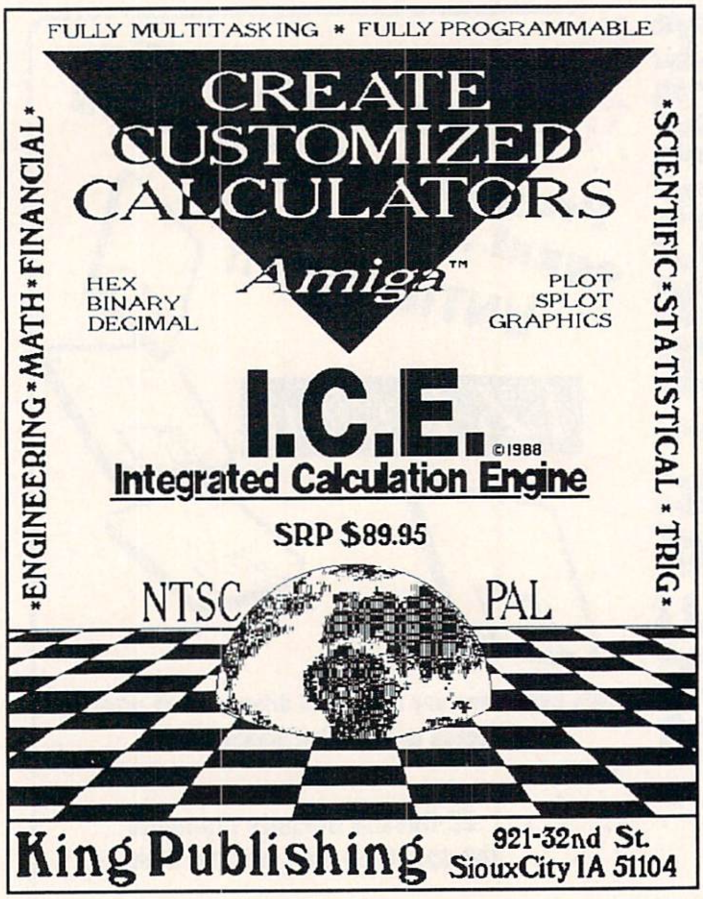
EZ-Backup provides easy recovery of individual files--
A simple to use mouse-oriented program allows you to recover individual files.

Works with all Amiga-DOS compatible hard drives.
(Amiga-DOS version 1.2 or higher)

- ⋆ Provides archive-bit utilities
- ⋆ Not copy protected
- ⋆ Multi-tasking
- ⋆ Complete manual with examples
- ⋆ Upgrades provided free for the first six months after program purchase!
- ⋆ Free telephone support!

Not more expensive just the best - $49.95 - from:
EZ-SOFT or an Amiga Dealer near you.
21125 Chatsworth Street
Chatsworth, CA 91311
(818) 341-8681
Dealer Inquires Welcome

EZ SOFT


FULLY MULTITASKING * FULLY PROGRAMMABLE
CREATE CUSTOMIZED CALCULATORS
Amiga™
HEX
BINARY
DECIMAL
PLOT
SPLOT
GRAPHICS
*ENGINEERING*MATH*FINANCIAL*
*SCIENTIFIC*STATISTICAL * TRIG*
I.C.E. ©1988
Integrated Calculation Engine
SRP $89.95
NTSC PAL
King Publishing
921-32nd St.
SiouxCity IA 51104

of those nifty utilities came from anyway?) Environment variables allow you to establish some general operating features of the command environment. For the ARP commands these variables include:

```
DATEFormat = 0        - default Amiga format:  DD-MMM-YY
           = 1        - International format:  YY-MM-DD
           = 2        - USA format:  MM-DD-YY
           = 3        - Canadian format:  DD-MM-YY
```

Every command that includes a date display will use the DATEFormat variable to determine how the date is displayed.

```
CopyFlags = C        - preserve the original date stamp
          = N        - copy the FileNote
          = rweda    - Protection bits: Disallow
          = RWEDA    - Protection bits: Allow
```

This variable will customize the COPY command actions and may be overridden with the FLAGS parameter of the COPY command. An example of the usage is shown below.

ESCAPE = \ - changes the Escape character (as used by ECHO etc.) to the backslash (the default character is *). You may set this to the character you prefer to use as the Escape character.

ARP commands recognize the following characters as special characters when preceded by the defined Escape character:

N - Newline
T - Horizontal Tab
V - Vertical Tab
B - Backspace
R - Return
F - Form Feed
E - ESCape (ASCII 27)
Xnn - Hex value nn

With the exception of the new ECHO command, these characters will only be recognized when placed between quotes.

Each of these variables may be changed with the SET command. To establish a variable, simply provide SET with the variable name and a value to set it to. For example:

```
1> SET DateFormat=1 CopyFlags=NCwd ESCAPE \
```

You could also establish some of your own variables with the SET command. The ones shown are those recognized by ARP commands. If you find that some of the ARP commands have problems with old script files that use the asterisk as the current window (such as the AmigaDOS COPY command) SET BCPL TRUE causes those ARP commands to not use the asterisk as a wildcard character. To display what all the current environment variables are and what their values are use:

```
1> SET LISTALL
```

I covered ASK,the other new commandin detail last issue. As you have seen, there are some very nice features provided by the ARP command collection. I have been using them for a while now and find that I definitely prefer them to the original AmigaDOS versions. You will find the ARP collection on Fred Fish Disk #123 or you can contact Microsmiths Inc. for order information at:

Arp Support
c/o *Microsmiths, Inc*
P.O. Box 561
Cambridge, MA 02140

•AC•

Death of a Process

The Polite Multitasking Guide to Error Handling

...in Modula-2

by Mark Cashman

Somewhere, deep in the hierarchy of modules and procedures, something has gone wrong. A memory allocation has failed, a task could not be spawned—an Amiga program falters, HALTs. But what about its resources: its screens, windows, and memory? Are they returned to the system, or left claimed until the next warm reboot?

That depends on your error handling strategy.

Most programming examples, including those in the ROM Kernel Manuals, and those in Amiga programming textbooks, give short shrift to error handling. This is necessary to cover the topic at hand. But in real life, if resources have already been allocated it is not enough to call Exit to terminate. The resources must be freed.

The Modula-2 language allows the programmer to write the various sections of program with the greatest possible independence. This independence simplifies a program's design. However, handling the type of errors that can arise during program execution can significantly complicate program logic in any language.

For instance, observe the following Modula-2 main module:

```
IF OpenedLibraries() THEN
 IF OpenedWindow() THEN
  IF AllocatedImage() THEN
   IF OpenedFile() THEN
    IF ReadImage() THEN
     DisplayImage;
    ELSE
     Error("Unable to read image.");
    END;
   ELSE
    Error("Unable to open file.");
   END
   CloseFile;
  ELSE
   Error("Unable to allocate image memory.");
  END;
  FreeImage;
 ELSE
  Error("Unable to open window.");
 END;
 CloseWindow;
ELSE
 Error("Unable to open libraries.");
END;
```

Clearly, the code is verbose. After all, concentrating on the normal case, we'd really prefer a main module that looked like this:

```
OpenLibraries;
OpenWindow;
AllocateImage;
ReadImage;
DisplayImage;
FreeImage;
CloseWindow;
CloseLibraries;
```

The example program solves this problem. The example is written in Modula-2 using the TDI libraries.

The example relies on some properties of the Modula language — its MODULE program unit, and its PROCEDURE variable. It also uses the Exec List and Node structures in the implementation.

(continued)

MODULEs are the most important construct in Modula. There are three types of modules in Modula:

Program modules
Executable programs — clients of library modules

Local modules
Modules local to a program module, library module, or PROCEDURE

Library modules
which have separately compilable DEFINITION and IMPLEMENTATION parts

Regardless of the module type, a MODULE always has the following characteristics:

Defines
Variable/constant types, constants, variables, procedures

Imports
Externally defined variable/constant types, constants, variables, procedures

Exports
Internally defined variable/constant types, constants, variables, procedures

Controls access to
Internally defined variable/constant types, constants, variables, procedures

Retains

Values of any module local variables as long as the surrounding scope lasts. Library modules retain their local variable values for the duration of the program run, as do modules local to a program or library module. Procedure local modules retain their local variable values as long as the procedure is active.

The following program module shows how variables in a module retain their values from one procedure call to another:

```
MODULE ShowVariableValuePersistence;
  IMPORT
    InOut;
  MODULE Accumulator;
    IMPORT
      InOut;
    EXPORT QUALIFIED
      Display,
      Increment;
    VAR
      Accumulator:              CARDINAL;
    PROCEDURE Increment(Cardinal: CARDINAL);
      BEGIN
        INC(AccumulatorVariable,Cardinal);
      END Increment;
    PROCEDURE Display;
      BEGIN
        InOut.WriteString("Accumulator = ");
        InOut.WriteCard(AccumulatorVariable,5);
InOut.WriteLn;
      END Display;
    BEGIN
      AccumulatorVariable:= 0;
    END Accumulator;
  BEGIN
    Accumulator.Increment(5);
    Accumulator.Display;
    Accumulator.Increment(5);
    Accumulator.Display;
  END ShowVariableValuePersistence.
```

When run, this module displays 5, then 10, showing how the AccumulatorVariable retains its value between calls to Increment. Of course, library modules also have this property.

A library module has a "client" that uses its services. The client may be another library module, a program module, or both.

The PROCEDURE variable allows the programmer to define a variable in which a procedure's address can be stored. It preserves the strict typing required by Modula by defining the expected parameters of a procedure whose address will be stored in the variable. For instance:

```
MODULE ProcedureVarDemo;

  TYPE
    ProcedureTYPE =
      PROCEDURE(CARDINAL,BOOLEAN): BOOLEAN
        (* Defines a procedure with a CARDINAL and a
           BOOLEAN parameter. The procedure also must
           return a BOOLEAN value *);
  VAR
    MyProcedureVariable:
      ProcedureTYPE;
  PROCEDURE OKProcedure
    (Cardinal: CARDINAL; Boolean: BOOLEAN): BOOLEAN;
    BEGIN
      RETURN TRUE;
    END OKProcedure;
  BEGIN
    MyProcedureVariable:= OKProcedure;
  END ProcedureVarDemo.
```

is a valid program. But this next one is not, because the parameter list of WrongTypeProcedure does not match that declared in the ProcedureTYPE definition:

```
MODULE ProcedureVarDemo;
  TYPE
    ProcedureTYPE =
      PROCEDURE(CARDINAL,BOOLEAN): BOOLEAN;
  VAR
    MyProcedureVariable:
      ProcedureTYPE;
  PROCEDURE WrongTypeProcedure
    (Cardinal: REAL; Boolean: BOOLEAN): BOOLEAN;
    BEGIN
      RETURN TRUE;
    END WrongTypeProcedure;
  BEGIN
    MyProcedureVariable:= WrongTypeProcedure;
  END ProcedureVarDemo.
```

Note that when a procedure is assigned to a procedure variable the parameter list is not specified.

Any procedure variable can be used as a procedure call. To extend the tiny example above:

```
MODULE ProcedureVarDemo;
  TYPE
    ProcedureTYPE =
      PROCEDURE(CARDINAL,BOOLEAN): BOOLEAN;
  VAR
    Boolean:
      BOOLEAN;
    MyProcedureVariable:
      ProcedureTYPE;

  PROCEDURE OKProcedure
    (Cardinal: CARDINAL; Boolean: BOOLEAN): BOOLEAN;
    BEGIN
      RETURN TRUE;
    END OKProcedure;
  BEGIN
    MyProcedureVariable:= OKProcedure;
    Boolean:= MyProcedureVariable(5,TRUE);
  END ProcedureVarDemo.
```

Most Modula program modules use the services of many library modules. Some of these are used indirectly — the program module calls upon a procedure in a library module that in turn calls upon a procedure from another library module, and so forth. Programmers frequently create library modules, both general and application dependent. Handling errors in these various modules, or freeing the resources allocated by various modules when an error causes termination, requires explicitly calling a procedure in each library module that is responsible for termination.

If such a policy is followed, which module calls the termination procedures? The module where the error occurs? Such a module might be able to call upon the termination procedures of the library modules it uses — but how can it call the termination procedures of library modules used only by the program module? And how do you prevent two library modules that use the same library module from calling that module's termination procedure twice?

You will see how to use a library module to keep track of the termination procedures of any set of modules, and how it guarantees the reclamation of all resources allocated by those modules.

The example consists of two modules — the library module TerminationService and the program module TestTerminationService.

The following is the definition of TerminationService:

```
DEFINITION MODULE TerminationService;

   TYPE

     ProcedureTYPE =
       PROCEDURE;

   PROCEDURE RegisterProcedure
     (TerminationProcedure: ProcedureTYPE);

   PROCEDURE Assert
     (Condition:         BOOLEAN;
      ModuleName:        ARRAY OF CHAR;
      ProcedureName:     ARRAY OF CHAR;
      ConditionMessage: ARRAY OF CHAR);

   PROCEDURE NormalTermination;

END TerminationService.
```

As you can see, the module defines a type — the TerminationService.ProcedureTYPE — and three procedures. RegisterProcedure accepts a ProcedureTYPE expression and adds it to a list of termination procedures kept internally by TerminationService. Assert tests that Condition is TRUE. If it is not TRUE, it calls the termination procedures in the order they were added to the list, tries to display the ConditionMessage, and terminates the program with an error code. (Note that the term "Assert" comes from a construct popular in programming language design a few years ago). NormalTermination should be called at the end of the program module — in fact, it must be called there if all resources are to be returned to the system at the end of a program run. NormalTermination will call all of the termination procedures (just like a failed Assert), but will RETURN to the caller without either a message or a DOS termination. For more details we have to turn to the IMPLEMENTATION part of this library module.

```
IMPLEMENTATION MODULE TerminationService;

   IMPORT

     DOSFiles,
     DOSLibrary,
     DOSProcessHandler,
     InOut,
     Intuition,
     Libraries,
     Lists,
     ListService,
     Memory,
     Nodes,
     SYSTEM;

   TYPE

     (* These types are private to this module. The
        types declared in the definition module are
        also available here. *)

     ProcedureNodeTYPE =
       RECORD
         Node: Nodes.Node;
         TerminationProcedure: ProcedureTYPE;
       END;
```

(continued)

```
  ProcedureNodePtrTYPE =
    POINTER TO ProcedureNodeTYPE;

VAR

  (* The values of these remain as long as the
     program module is running *)

  ProcedureList:
    Lists.List;

  RememberPtr:
    Intuition.RememberPtr;

PROCEDURE RegisterProcedure

  (TerminationProcedure: ProcedureTYPE);

  VAR

    ProcedureNodePtr:
      ProcedureNodePtrTYPE;

  PROCEDURE AllocateProcedureNodePtr():

     ProcedureNodePtrTYPE;

    VAR

      Ptr:
        SYSTEM.ADDRESS;

    BEGIN

      Ptr:=
        Intuition.AllocRemember
          (RememberPtr,
           SYSTEM.TSIZE(ProcedureNodeTYPE),
           Memory.MemReqSet{Memory.MemClear});

      Assert
        (Ptr # SYSTEM.NULL,
         "TerminationService",
         "RegisterProcedure",
         "Allocated procedure node");

      RETURN Ptr;

    END AllocateProcedureNodePtr;

  BEGIN

    ProcedureNodePtr:= AllocateProcedureNodePtr();

    ProcedureNodePtr^.TerminationProcedure:=
      TerminationProcedure;

    (* Note: Procedures are called in FIFO order *)

    Lists.AddTail
      (Lists.ListPtr(SYSTEM.ADR(ProcedureList)),
       Nodes.NodePtr(ProcedureNodePtr));

  END RegisterProcedure;
```

```
PROCEDURE CallRegisteredProcedure

    (NodePtr: Nodes.NodePtr);

  VAR

    ProcedureNodePtr:
      ProcedureNodePtrTYPE;

  BEGIN

    ProcedureNodePtr:=
      ProcedureNodePtrTYPE(NodePtr);

    ProcedureNodePtr^.TerminationProcedure;

  END CallRegisteredProcedure;

PROCEDURE CallRegisteredProcedures;
  BEGIN

    ListService.TraverseList
      (Lists.ListPtr(SYSTEM.ADR(ProcedureList)),
       CallRegisteredProcedure);

  END CallRegisteredProcedures;

PROCEDURE Assert
  (Condition:          BOOLEAN;
   ModuleName:         ARRAY OF CHAR;
   ProcedureName:      ARRAY OF CHAR;
   ConditionMessage: ARRAY OF CHAR);

  BEGIN

    IF Condition THEN

      RETURN;

    ELSE

      IF

        DOSFiles.IsInteractive(DOSFiles.Output())

      THEN
        InOut.WriteString("Assert failed: ");
        InOut.WriteLn;
        InOut.WriteString("  Module     - ");
        InOut.WriteString(ModuleName);
        InOut.WriteLn;
        InOut.WriteString("  Procedure - ");
        InOut.WriteString(ProcedureName);
        InOut.WriteLn;
        InOut.WriteString("  Assertion - ");
        InOut.WriteString(ConditionMessage);
        InOut.WriteLn;
      END;

      CallRegisteredProcedures;

      IF RememberPtr # SYSTEM.NULL THEN
        Intuition.FreeRemember(RememberPtr,TRUE);
      END;

      Libraries.CloseLibrary
        (Intuition.IntuitionBase);

      (* Can't close DOS before using one of its
         procedures, so no CloseLibrary
          (DOSLibrary.DOSBase) here *)

      DOSProcessHandler.Exit(20);

    END;

  END Assert;
```

```
PROCEDURE NormalTermination;
  BEGIN

    CallRegisteredProcedures;

    IF RememberPtr # SYSTEM.NULL THEN
      Intuition.FreeRemember(RememberPtr,TRUE);
    END;

    Libraries.CloseLibrary(Intuition.IntuitionBase);
    Libraries.CloseLibrary(DOSLibrary.DOSBase);

  END NormalTermination;

BEGIN (* Module initialization *)

  Lists.NewList
    (Lists.ListPtr(SYSTEM.ADR(ProcedureList)));

  RememberPtr:= SYSTEM.NULL;

  DOSLibrary.DOSBase:=
    Libraries.OpenLibrary(DOSLibrary.DOSName,0);

  IF DOSLibrary.DOSBase = SYSTEM.NULL THEN
    (* Can't use unopened DOS to Exit... *)
    HALT;
  END;

  Intuition.IntuitionBase:=
    Libraries.OpenLibrary
      (Intuition.IntuitionName,0);

  IF Intuition.IntuitionBase = SYSTEM.NULL
    THEN DOSProcessHandler.Exit(21); END;

END TerminationService.
```

As you can see, TerminationService has its own module initialization section. In a program module, this would correspond to the C language main() function. In a library module, the initialization section sets initial values for internal variables. Since the initialization section of a library module is performed before the initialization section of any module importing from it, it is guaranteed that TerminationService will be initialized before any of its clients can call upon it.

The initialization section sets up a list that will be used to record the termination procedures passed to RegisterProcedure. This list is a standard Exec List, with an extended Node structure for each list element; the extension stores the value of the procedure variable passed to RegisterProcedure. The RememberPtr, used in conjunction with the Intuition library calls AllocRemember and FreeRemember, is initialized to SYSTEM.NULL to signal that it is empty. Later, it will contain a list of memory allocated by TerminationService procedures. This memory will be the Nodes of the ProcedureList.

The initialization section also opens the Amiga libraries (which are not the same as library modules) for Intuition (used to implement the AllocRemember and FreeRemember procedures) and the DOS (used by the Exit procedure). This frees the program module from needing to open these libraries, and means the program module need not know that TerminationService uses either of these libraries.

The RegisterProcedure procedure creates a ProcedureNodeTYPE area of memory, and sets the TerminationProcedure part of the ProcedureNode to the value passed by the client. It then links the properly initialized node to the tail of the list. Note that, if the allocation of the list node fails, the Assert procedure is used to make sure that the various registered termination procedures are called before Exit. This is how you would use it in your own library module. Also note that Assert is called from a position in the program before its definition. Modula allows this because the Assert procedure is declared in the DEFINITION MODULE. Normally, everything in Modula must be declared before it is used.

The CallRegisteredProcedures procedure uses a list scanning procedure from my module ListService. The implementation of this procedure (and those it calls) is given below:

```
PROCEDURE AtEndOfList

  (ListPtr:        Lists.ListPtr;
   CurrentNodePtr: Nodes.NodePtr):

    BOOLEAN;

  BEGIN

    RETURN (* TRUE IF *)
      CurrentNodePtr^.lnSucc = SYSTEM.NULL;

  END AtEndOfList;

PROCEDURE NextNode

  (CurrentNodePtr: Nodes.NodePtr): Nodes.NodePtr;

  BEGIN

    RETURN CurrentNodePtr^.lnSucc;

  END NextNode;

PROCEDURE TraverseList

  (ListPtr:
     Lists.ListPtr;

   ListTraversalStepRoutine:
     ListTraversalStepRoutineTYPE);

  VAR

    NodePtr:
      Nodes.NodePtr;

  BEGIN

    NodePtr:= ListPtr^.lhHead;

    WHILE NOT AtEndOfList(ListPtr,NodePtr) DO

      ListTraversalStepRoutine(NodePtr);
      NodePtr:= NextNode(NodePtr);

    END;

  END TraverseList;
```

(continued)

The Bit Bucket
COMPUTER STORE

We Want Your Business!!
We Have the Best Prices!!
Oldest Commodore Dealer in the Area!!

2 Locations to Serve You

- Software
- Hardware
- Service
- Information

1294 Washington Street
W. Newton MA 02165
617-964-3080

621 Boston Post Road
Sudbury MA 01776
508-443-9731

COMMODORE AMIGA™
Authorized Dealer

Authorized Commodore Amiga Dealer and Commodore Service Center

Note that the procedure variable is used to generalize the TraverseList procedure, which can simply implement scanning of the nodes. This helps keep the ListTraversalStepRoutine simple.

The last module is a test program module that tries out some of the TerminationService module features:

```
MODULE TestTerminationService;

  IMPORT

    InOut,
    Strings,
    TerminationService;

  FROM TerminationService IMPORT Assert;

  VAR

    Count:
      CARDINAL;

  PROCEDURE OK(Message: ARRAY OF CHAR);
    BEGIN

      InOut.WriteString(Message); InOut.WriteLn;

    END OK;

  PROCEDURE Test;
    BEGIN

      OK("Bye Bye...");
      InOut.WriteCard(Count,5); InOut.WriteLn;
      INC(Count);

    END Test;

  BEGIN

    Count:= 0;

    OK("Register procedure");

    TerminationService.RegisterProcedure(Test);
    TerminationService.RegisterProcedure(Test);
    TerminationService.RegisterProcedure(Test);
    TerminationService.RegisterProcedure(Test);

    InOut.WriteString
      ("Press RETURN to fail, anything else to ");
    InOut.WriteString("continue...");
    InOut.ReadString(Input);

    OK("Assert");

    Assert
      (Strings.Length(Input) # 0,
       "TestTerminationService",
       "Program module initialization",
       "String entered");

    OK("Normal termination");

    TerminationService.NormalTermination;

END TestTerminationService.
```

This program displays a sequence number for every call to the termination procedure. Since the termination procedure was registered five times, the program will print the numbers from 0 through 4. Your entry to the prompt determines whether the Assert will fail, or whether NormalTermination will occur.

There were no major problems in developing this module. In fact, it was up and working within one evening. Yet it will save me hundreds of lines of error handling code over the next year. An inexpensive investment for a large payoff.

The hardest thing to understand is the PROCEDURE type variable— that it can be a variable and a procedure at the same time; how to call it; and that your call parameter list must match the requirements. The next hardest thing to understand is that module variables keep their values even between using procedures from the module. The third hardest thing to understand is setting up, adding to, and traversing the list structure.

For more information on modules and procedure variables, see Niklaus Wirth's Programming In Modula-2 (1983).

That's all folks!

One final note — according to Wirth's Programming in Modula-2 (1983), procedures assigned to procedure variables must not be nested within another procedure — they must be global to the module. It cannot be guaranteed what will happen if you violate this.

I have not yet tested this module with a Workbench activated program, which would be interesting. Maybe you can try it and let me know.

•AC•

INDUSTRIAL STRENGTH
MENUS

Add Some Snazzy SubMenus To Your AmigaBASIC Cuisine

by Robert D'Asto

The ability to quickly and easily create practical menus is a terrific feature of AmigaBASIC. We can lay out a rather extensive bill of fare in very short order using the powerful MENU statement and link the offered selections to your program as neat as you please with the MENU(0) and MENU(1) functions. Up to 10 separate menus, each with up to 20 items can be produced in this way, an ample number of choices for nearly any application. We can thank not only the language, but that section of the operating system known as Intuition for this useful feature. It is Intuition which supplies all the needed "building materials" which BASIC then assembles to our specifications in workmanlike fashion via simple MENU statements a much simpler state of affairs than the C programmer faces when constructing menus.

That's the good news. The not so good news is that our creative menu craft seems somewhat limited in scope in comparison to that of the Cjockey. The latter could nail up as many as 1,953 selections, arranged in both menus and submenus, and all with a wide array of color and display options to boot! Kind of leaves BASIC in the dust, doesn't it?

Wait a second...AmigaBASIC is no wimp! Our beloved language is capable of producing professional menu features quite beyond those described in the "factory manual" including submenus, color options and others. We simply have to "bypass the middle man" somewhat and deal directly with some of the inner regions of the Amiga operating system itself. There are some new things that need learning, to be sure, but I think you'll find it's well worth the effort as these new concepts can put your BASIC programming on a whole new level of expertise, as well as greatly enhancing your understanding and enjoyment of the Amiga.

Enter the SubMenu_Demo listed at the end of this article into your BASIC editor. The listing contains two LIBRARY calls, "intuition.library" and "exec.library", so the appropriate bmap files must be available. The code, as written, assumes that the intuition.bmap and exec.bmap files exist either in the "libs:" directory of the WorkBench disk or in the same subdirectory as the program itself. If you prefer another arrangement you'll need to adjust the LIBRARY statements accordingly.

Take your time entering the code and be sure to save the program before trying it out. Errors can be more difficult to track down when dealing with library routines and POKE statements as a goof is more likely to cause a system crash, requiring a reboot, rather then simply an error message. It's easier to just go slowly and double check your code as you type.

Now, run the program and let's take a look at what we've wrought. Holding down the right mouse button should reveal two menus, each with a number of item selections and...gadzooks!...SUBITEMS!!...and our program can even detect which subitem's been chosen!!!...think of the possibilities!!!!

All humbleness aside, what we have here is a collection of subprograms which can be used with any BASIC program to generate submenus containing up to 10 selections each, and then link them to any existing menu item we wish. This now gives us a potential of 10 menus, each with 20 items, and up to 10 subitems per item, or a theoretical total of up to 2,000 menu selections! To tell you the truth, I haven't really tested out this maximum limit, but it certainly does a dandy job of adding all the submenus I've needed so far. It also adds some visual pizazz to that ordinary, à la carte menu, doesn't it?...and it gives the user interface added utility and a more professional look in the process. We can even tweak a parameter here and there and add some other options which we'll get to in a moment.

As you've probably noticed, the program accomplishes its submenu tasks almost entirely through the use of subprograms. This was intentional. Once you've entered the code into the editor you can simply save all the subprograms with the ASCII option (SAVE "filename",A) and then MERGE these subs into any future program to which you wish to add SubMenus. The subs were also designed for ease of use and to parallel the MENU statements as much as possible. These subs will be explained in more detail later on...first, let's backtrack a bit and explain the "how" and the "why" of them.

Programming Mysteries Revealed

You may have seen programs like this elsewhere, containing PEEKs and POKEs, and wondered how the heck the writer ever figured out where to PEEK and what to POKE in order to produce some arcane feat of Amiga wizardry. It's not all that mysterious, really. If you've never used the PEEK and POKE commands before, take a moment to look them up in your AmigaBASIC manual and familiarize yourself with them and their variations. They're simply used to read or write values from or to specific locations in RAM. Each has three variations: one each for 1-byte, 2-byte or 4-byte values.

Using these two commands for advanced Amiga programming requires some familiarity with the Amiga operating system and an understanding of a particular programming concept with which many BASIC programmers are unfamiliar. This concept is called "structures" and is a very important one to

(continued)

the Amiga because its system is absolutely loaded with them and any "deep" programming of the machine requires a grasp of what they are and how to use them. Unfortunately, discussions of structures are usually omitted from AmigaBASIC tutorials so documentation of "deep AmigaBASIC" procedures is hard to find. Some say that AmigaBASIC doesn't support structures...well, not directly perhaps, but we can still support them "indirectly" just fine and thus add a vast array of new tricks up our BASIC sleeves. Perhaps the rumored 1.4 upgrade of AmigaBASIC will include some improved support of structures but, in the mean time, we can do handily with what we have. It just takes a little homework to get a grasp of what structures are and how to use them.

A structure is simply a type of organization of data. If you understand variable arrays you're about half way home to an understanding of structures. An array is a named group of variables. Each variable in the array is assigned a number (called a subscript) and can then be referred to by stating the array name and its number, like this: BoomArray(17), which means the 18th variable (we start counting from zero) in an array called BoomArray. One limitation of arrays is that every variable they contain must be of the same type: Long integer, short integer, single float, double float, or string. A point worth noting is that all the array elements occupy a single, continuous block of memory arranged in the sequence of their subscripts. The word normally used to describe this type of memory storage is "contiguous", which just means: Arranged in a sequence, each touching the next with no spaces in between. Imagine a Los Angeles freeway with rush hour, bumper-to-bumper traffic and you'll know what contiguous means.

A structure is also a group of variables but it differs from an array in that it can contain any mixture of different variable types. That's really the most important aspect of a structure. Let's say we create a mixed bag of values like this:

```
1641022
3741729
314
1227
61
```

In this example structure we have a sequence of two long integers (requiring 4 bytes each), two short integers (2 bytes each) and our fifth "element" is a number requiring only one byte of memory space. This is perfectly acceptable as a structure. The "elements" of a structure, by the way, are called "fields" and, like an array, the structure's fields occupy a contiguous block of memory. In this case, we'd have a sequence of "4 bytes...4 bytes...2 bytes...2 bytes...1 byte" to hold our structure. One might ask why we don't just use a long integer array to hold all the values and forget about structures. We could do that but it would result in a somewhat wasteful result memory wise. We'd end up using 4 bytes multiplied by the five fields or 20 bytes of memory if we used an array, contrasted by only 13 bytes using a structure. AmigaBASIC arrays also have a certain propensity for shifting around to various memory locations during a run and there's no easy way to keep them nailed down for the kind of operations necessary here. Structures, though, don't mind being assigned a permanent address which makes them much simpler to locate when needed.

There are basically four things that a programmer can do with a structure: Create one, "uncreate" one, read a value within a field in an existing structure, or change the field in an existing structure. Some structures are "automatically" created by the operating system and some are created solely by the program. For example, when we create a window within our programs with the WINDOW statement, a Window Structure is created in RAM which contains all of its parameters and other necessary information needed by the system to interact with that window. We didn't create that structure directly (though we could), but we can gain access to it and make desired changes in the window by changing values in its Window Structure. A structure, then, can be used as a sort of "control panel" for various system objects. Thus, the more you know about structures, the more control you have of the system.

Creating a structure with AmigaBASIC could simply be done with a sequence of POKE statements, but there's one problem with this approach: Where do we put it? If we just choose a memory address at random and start POKEing away we run the risk of destroying some needful data that already existed in that particular area of RAM. We need a way to find an "unoccupied" area in memory of the correct size first. Fortunately there is a System Routine which specializes in just this exact "service". It's an "exec.library" function called AllocMem and its function is to allocate memory requested by the program and once it has done this it returns the memory address of the area so allocated. The programmer can then simply POKE the desired fields, beginning at the address provided by AllocMem. There is also another exec routine called FreeMem which does the opposite: We "feed" it a particular memory address and the length (in bytes) and it returns that part of RAM to the status of free memory. These two routines are easy to use and are described in more detail below.

Changing a field in an existing structure is also quite easy. All we need is the structure's starting address. For example, if we wanted to change the "314" in the above structure to "315" we could do it like this:

```
POKEW StartingAddress& + 8, 315.
```

where "StartingAddress&" is the memory address of the first byte of the structure. Since the field containing the "314" begins 8 bytes beyond the beginning of the structure we just add 8 to the structure's address and POKEW in the new 2cbyte number. Reading a field in a structure is done the same way. If we program:

```
PRINT PEEKW (StartingAddress& + 8)
```

we'll now get the "315" that we POKEWed earlier.

Getting a handle on the above four basic actions with structures opens the door to a great many "deep" or "low level" programming activities for AmigaBASIC and puts it in the same development language ball park as C. The Amiga makes extensive use of structures in delineating various objects and constructs which make up its operating system and user interface. The better your understanding and ability to control these structures, the more you will be able to create your own brand of magic with the Amiga.

The Structure Plot Thickens

A menu display is simply a network of structures. These are of two types: Menu Structures and MenuItem Structures. The arrangement and explanation of these structure follows but don't concern yourself with memorizing this information or trying to absorb it all at once. It's here for reference as needed and will become more familiar through use. The following two structures are used by the operating system to "describe" a Menu (a single category of selections) and a MenuItem, that is, each contains all the information needed by the system to display, use and interact with these objects. A Menu Structure must exist for each Menu (category of Items) and every Item must have a MenuItem Structure. They look like this:

Menu Structure:

field #1—4 bytes containing the address of next Menu Struct
field #2—four 2-byte numbers: LeftEdge,TopEdge,Width,Height
field #3—a 2-byte number called Flags
field #4—4 bytes containing address of MenuName
field #5—4 bytes containing address of 1st MenuItem Struct

MenuItem Structure:

field #1—4 bytes containing address of next MenuItem Struct
field #2—four 2-byte numbers: LeftEdge,TopEdge,Width,Height
field #3—a 2-byte number called Flags
field #4—a 4-byte number called MutualExclude
field #5—4 bytes containing address of Menu Item text
field #6—6 bytes called Command
field #7—4 bytes containing address of SubItem Struct
field #8—a 2-byte number called NextSelect

First I'll explain the Menu Structure. The first of its fields is self-explanatory. It's just the address in memory of another Menu Structure which defines the next Menu. The last Menu would have a "0" in this field. When a structure contains the address of another structure this is called "linking" and all the Menu Structures so linked are referred to as a "linked list". A memory address used for linking is referred to as a "pointer". Moving to field #2 we see 4 numbers of 2 bytes each, for a total of 8 bytes within this one field. The first two numbers give the screen location of the box containing the Menu title and the last two give its dimensions. The 3rd field, Flags, describes certain display options and will be discussed later. The fourth field is a "pointer" to (memory address of) another structure which describes the text of the Menu title, and the last field contains the memory address of the first MenuItem Structure in that Menu.

The second structure above is the MenuItem Structure. Most of its fields are analogous to those in the Menu Structure, but there are a few new ones. The MutualExclude field (#4) contains a number used to describe which, if any, Menu Items will be "ghosted" when this Item is selected by the user. Field #5 contains a pointer to a text string describing what, if any, keyboard key can be pressed along with the RIGHT Amiga key as an alternate method of selecting this item. Field #7 is of special importance here as it contains the address of the first SubItem Structure associated with this MenuItem. What does this SubItem Structure look like? Its identical to a MenuItem Structure and the first field of each is linked to the next in the same manner as MenuItem Structures. The last field above contains a number called NextSelect. Intuition provides this

(continued on page 83)

Would you use your Amiga like this?

Without ENCORE you might as well be!

Encore allows you to build macros by recording mouse movements and keystrokes. You can assign each macro to a key combination so that with a touch of the finger any macro can instantly be played back.

Macros can have spoken narration.

Macros can have scrolling text at the bottom of the monitor.

Macros can loop back for repeated playback.

Macros can be either relative or absolute.

Macros can be appended.

Macros can be user interactive.

Macros can be time or file compressed.

Macros can be of unlimited size.

Automatic mouse and keyboard lockout.

Works with any Amiga program.

Speech preferences program with phonemized word database.

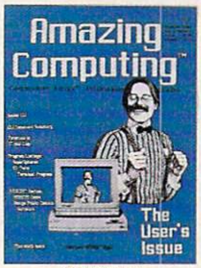
No matter what you are using your Amiga for, Encore is a must for your software library. See your local dealer for details.

ENCORE is NOT COPY PROTECTED Suggested retail $69.95
Dealer inquiries invited.

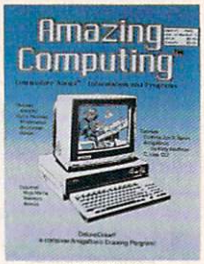
ELIPSYS **PO Box 16757**
Seattle, Wa 98116
(206)938-0825
FAX (206)932-9520

Amazing COMPUTING™

• EXPANDING REFERENCE •


VOLUME 1.1

VOLUME 1.2

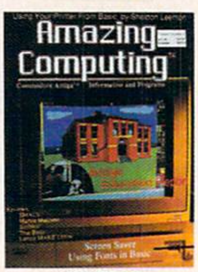

VOLUME 1.3

VOLUME 1.4

VOLUME 1.5

VOLUME 1.6

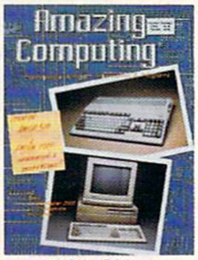
VOLUME 1.7

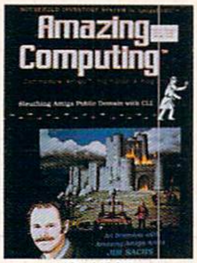

VOLUME 1.8

VOLUME 1.9

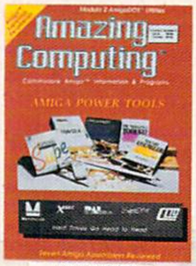
VOLUME 2.1

VOLUME 2.2


VOLUME 2.3

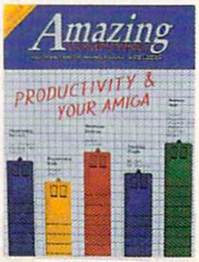

VOLUME 2.4

VOLUME 2.5

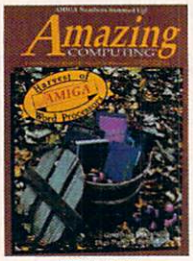

VOLUME 2.6

VOLUME 2.7

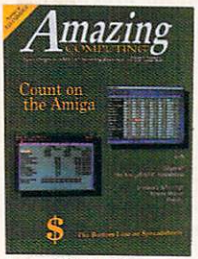
VOLUME 2.8

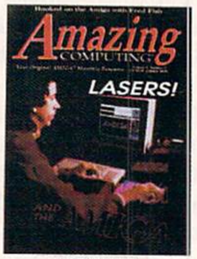

VOLUME 2.9

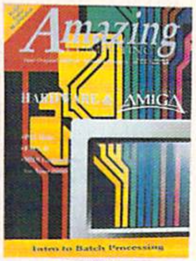
VOLUME 2.10


VOLUME 2.11

VOLUME 2.12


VOLUME 3.1

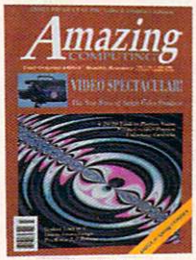

VOLUME 3.2


VOLUME 3.3

VOLUME 3.4

VOLUME 3.5

VOLUME 3.6


VOLUME 3.7

VOLUME 3.8

VOLUME 3.9

Your Resource to the Commodore Amiga™

Expanding reference is not just an empty promise. The pages of Amazing Computing™ are filled with articles on technical operations and procedures, basic use, and just-plain-fun. The growing library of Amazing Computing's Back Issues contains articles ranging from building your own IBM Disk controller, to setting up your own startup sequence. Amazing Computing™ has repeatedly been the first magazine to offer the Amiga users solid, in depth reviews and hands on articles for their machines.

From the Beginning

Since February 1986, Amazing Computing™ has been providing users with complete information for their Amigas. This store house of programs and information is still available through our back issues. From the Premiere issue to the present, there are insights into the Amiga any user will find useful. AC was the first magazine to document CLI, tell its readers how to connect a 5 1/4 IBM drive, describe a 1 meg upgrade hardware project for the A1000, and many more. Please read the list of topics AC has covered below to find the information you have been missing.

Back Issues are $5.00 US, $6.00 Canada and Mexico, $7.00 Foreign Surface

All payments must be made by check or money order in U.S. funds drawn on a U.S. Bank.

Limited Supply

Unfortunately, nothing lasts forever, and the availability of some of our Back Issues is definitely limited. Complete your Amazing Computing™ library today, while these issues are still available, by completing the order form in the back of this issue.

Volume 1 Number 1 Premiere 1986

Super Spheres By Kelly Kauffman An ABasic Graphics prog.
Date Virus By J Foust A disease may attack your Amiga!
EZ-Term by Kelly Kauffman An ABasic Terminal program
Miga Mania by P. Kivolowitz Programming fixes & mouse care
Inside CLI by G. Musser a guided insight into the AmigaDos™
CLI Summary by G. Musser Jr. A list of CLI commands
AmigaForum by B. Lubkin Visit Compuserve's Amiga SIG
Commodore Amiga Development Program by D. Hicks
Amiga Products A listing of present and expected products

Volume 1 Number 2 March 1986

Electronic Arts Comes Through A review of software from EA
Inside CLI: part two G. Musser Investigates CLI & ED
A Summary of ED Commands
Live! by Rich Miner A review of the Beta version of Live!
Online and the CTS Fabite 2424 ADH Modem by J. Foust
Superterm V 1.0 By K. Kauffman A term. prog. in Amiga Basic
A Workbench "More" Program by Rick Wirch
Amiga BBS numbers

Volume 1 Number 3 April 1986

Analyze! a review by Ernest Viverios
Reviews of Racter, Barataccas and Mindshadow
Forth! The first of our on-going tutorial
Deluxe Draw!! by R. Wirch An Amiga Basic art program
Amiga Basic, A beginners tutorial
Inside CLI: part 3 by George Musser George gives us PIPE

Volume 1 Number 4 May 1986

SkyFox and Artictox Reviewed
Build your own 5 1/4 Drive Connector By Ernest Viveiros
Amiga Basic Tips by Rich Wirch
Scrimper Part One by P. Kivolowitz prog to print Amiga screen
Microsoft CD ROM Conference by Jim O'Keane
Amiga BBS Numbers

Volume 1 Number 5 1986

The HSI to RGB Conversion Tool
by S. Pietrowicz Color manipulation in BASIC
AmigaNotes by Rick Rae The first of the Amiga music columns
Sidecar A First Look by John Foust A first "under the hood"
John Foust Talks with R. J. Mical at COMDEX™
How does Sidecar affect the Transformer
an interview with Douglas Wyman of Simile
The Commodore Layoffs by J. Foust A look Commodore "cuts"
Scrimper Part Two by Perry Kivolowitz
Marauder reviewed by Rick Wirch
Building Tools by Daniel Kary

Volume 1 Number 6 1986

Temple of Apshai Triology reviewd by Stephen Pietrowicz
The Halley Project: A Mission reviewed by S Pietrowicz
Flow: reviewed by Erv Bobo
Textcraft Plus a First Look by Joe Lowery
How to start your own Amiga User Group by William Simpson
Amiga User Groups
Mailing List by Kelly Kauffman a basic mail list program
Pointer Image Editor by Stephen Pietrowicz
Scrimper: part three by Perry Kivolowitz
Fun With the Amiga Disk Controller by Thom Sterling
Optimize Your AmigaBasic Programs for Speed by Pietrowicz

Volume 1 Number 7 1986

Aegis Draw: CAD comes to the Amiga by Kelly Adams
Try 3D by Jim Meadows an introduction to 3D graphics
Aegis Images/ Animator: a review by Erv Bobo
Deluxe Video Construction Set reviewed by Joe Lowery
Window requesters in Amiga Basic by Steve Michel
ROT by Colin French a 3D graphics editor
"I C What I Think" Ron Peterson with a few C graphic progs
Your Menu Sir! by B Catley program Amiga Basic menues
IFF Brush to AmigaBasic 'BOB' Basic editor by M Swinger
Linking C Programs with Assembler Routines...by Gerald Hull

Volume 1 Number 8 1986

The University Amiga By G.Gamble Amiga at Washington State
MicroEd a look at a one man army for the Amiga
MicroEd, The Lewis and Clark Expedition reviewed Frizelle
Scribble Version 2.0 a review
Computers in the Classroom by Robert Frizelle
Two for Study by Frizelle Discovery & TheTalking Coloring Book
True Basic reviewed by Brad Grier
Using your printer with the Amiga
Marble Madness reviewed by Stephen Pietrowicz
Using Fonts from AmigaBasic by Tim Jones
Screen SaVer by P. Kivolowitz A monitor protection prog. in C
Lattice MAKE Utility reviewed by Scott P. Evernden
A Tale of Three EMACS by Steve Poling
.bmap File Reader in Amiga Basic by T Jones

Volume 1 Number 9 1986

Instant Music Reviewed by Steve Pietrowicz
Mindwalker Reviewed by Richard Knepper
The Alegra Memory Board Reviewed by Rich Wirch
TxEd Reviewed by Jan and Cliff Kent
Amazing Directory A guide to the sources and resources
Amiga Developers A listing of Suppliers and Developers
Public Domain Catalog A listing of Amicus and Fred Fish PDS
Dos 2 Dos review R. Knepper Transfer files from PC/MS-DOS
MaxiPlan review by Richard Knepper The Amiga Spreadsheet
Gizmoz by reviewed by Peter Wayner Amiga extras!
The Loan Information Program by Brian Catley
basic prog. to for your financial options
Starting Your Own Amiga Related Business by W. Simpson
Keep Track of Your Business Usage for Taxes by J. Kummer
The Absoft Amiga Fortran Compiler reviewed by R A. Reale
Using Fonts from AmigaBasic, Part Two by Tim Jones
68000 Macros on the Amiga by G. Hull Advance your ability.
TDI Modla-2 Amiga Compiler review by S Faiwisze

Volume 2 Number 1 1987

What Digi-View Is... Or, What Genlock Should Be! by J. Foust
AmigaBasic Default Colors by Bryan Catley
AmigaBasic Titles by Bryan Catley
A Public Domain Modula-2 System reviewed by Warren Block
One Drive Compile by Douglas Lovell Lattice C with one drive
A Megabyte Without Megabucks by Chris Irving
An Internal Megabyte upgrade
Digi-View reviewed by Ed Jakober
Defender of the Crown reviewed by Keith Conforti
Leader Board reviewed by Chuck Raudonis
Roundhill Computer System's PANEL reviewed by Ray Lance
Digi-Paint.....by New Tek previewed by John Foust
Deluxe Paint II ....from Electronic Arts previewed by J. Foust

Volume 2 Number 2 1987

The Modem by Joseph L. Rothman efforts of a BBS Sysop
MacroModem reviewed by Stephen R. Pietrowicz
GEMINI or "It takes two to Tango" by Jim Meadows
Gaming between machines
BBS-PC! reviewed by Stephen R. Pietrowicz
The Trouble with Xmodem by Joseph L. Rothman
The ACO Project....Graphic Teleconferencing on the Amiga
by S. R. Pietrowicz
Flight Simulator II....A Cros Country Tutorial by John Rafferty
A Disk Librarian in AmigaBASIC by John Kennan
Creating and Using Amiga Workbench Icons by C. Hansel
AmigaDOS version 1.2 by Clifford Kent
The Amazing MIDI Interface build your own by Richard Rae
AmigaDOS Operating System Calls and
Disk File Management by D. Haynie
Working with the Workbench by Louis A. Mamakos Prog in C

Volume 2 Number 3

The Amiga 2000™ by J Foust
A First look at the new, high end Amiga™
The Amiga 500™ by John Foust
A look at the new, low priced Amiga
An Analysis of the New Amiga PCs by J. Foust
Speculation on the New Amigas
Gemini Part II by Jim Meadows
The concluding article on two-player games
Subscripts and Superscripts in AmigaBASIC by Ivan C. Smith
The Winter Consumer Electronics Show by John Foust
AmigaTrix by W. Block Amiga™ shortcuts
Intuition Gadgets by Harriet Maybeck Tolly
A journey through gadget-land, using C
Shanghai reviewed by Keith M. Conforti
Chessmaster 2000 & Chessmate reviewedby Edwin V. Apel, Jr.
Zing! from Meridian Software reviewed by Ed Bercovitz
Forth! by Jon Bryan Get stereo sound into your Forth programs.
Assembly Language on the Amiga™ by Chris Martin
Roomers by theBandito Genlocks are finally shipping, & MORE!!!
AmigaNotes by R. Rae Hum Busters... "No stereo? Y not?...
The AMICUS Network by J. Foust
CES, user group issues and Amiga Expo"

Volume 2 Number 4 1987

Amazing Interviews Jim Sachs by S. Hull Amiga Artist
The Mouse That Got Restored by Jerry Hull and Bob Rhode
Sluething Public Domain Disks with CLI by John Foust
Highlights: the San Francisco Commodore Show by S Hull
Speaker Sessions: San Francisco Commodore Show H Tolly
Household Inventory System in AmigaBASIC™ by B Catley
Secrets of Screen Dumps by Natkun Okun
Using Function Keys with MicroEmacs by Greg Douglas
Amigatrix II by Warren Block More Amiga shortcuts
Basic Gadgets by Brian Catley Create gadget functions
Gridiron reviewed by K. Conforti Real football for the Amiga
Star Fleet I Version 2.1 reviewed by J. Tracy Amigain Space
The TIC reviewed by J. Foust Battery powered Clock Calendar
Metascope review by H. Tolly An easy-to-use debugger

Volume 2 Number 5 1987

The Perfect Sound Digitizer review by R. Battle
The Future Sound Digitizer by W. Block Applied Vision's SD
Forth! by J. Bryancomparing JForth amd Multi-Forth.
Basic Input by B. Catley AmigaBASIC input routine for use in
all your programs.
Writing a SoundScape Module in C by T. Fay Programming with MIDI, Amiga and SoundScape by SoundScape author.

Volume 2 Number 5 1987 continued

Programming in 68000 Assembly Language by C. Martin
Continuing with Counters & Addressing Modes.
Using FutureSound with AmigaBASIC by J. Meadows
AmigaBASIC Programming utility with real, digitized STEREO
AmigaNotes Rich Rae reviews SoundScape Sound Sampler.
More AmigaNotes by R. Rae A furtherlook at Perfect Sound .
Waveform Workshop in AmigaBASIC by J. Shields edit & save
waveform for use in other AmigaBASIC programs.
The Mimetics Pro MIDI Studio by Sullivan, Jeffery
A review of Mimetics' music editor/player.
Intuition Gadgets Part II by H. MaybeckTolly Boolean gadgets
provide the user with an on/off user interface.

Volume 2 Number 6 1987

Forth! by J. Bryan Access resources in the ROM Kernal.
The Amazing Computing Hard Disk Review by J. Foust & S. Leemon In-depth looks at the C Ltd. Hard Drive, Microbotics' MAS-Drive20, Byte by Byte's PAL Jr., Supra's 4x4 Hard Drive and Xebec's 9720H Hard Drive. Also. a look at disk driver software currently under devlopment.

Modula-2 AmigaDOS™ Utilities by S. Faiwiszewsk A
Calls to AmigaDOS and the ROM kernal.
Amiga Expansion Peripheral by J. Foust
Explanation of Amiga expansion peripherals.
Amiga Technical Support by J. Foust
How and where to get Amiga tech support.
Goodbye Los Gatos by J. Foust Closing Los Gatos.
The Amicus Network by J. Foust West Coast Computer Faire.
Metacomco Shell and Toolkit by J. Foust A review
The Magic Sac by J. Foust Run Mac programs on your Amiga.
What You Should Know Before Choosing an Amiga 1000
Expansion Device by S. Grant
7 Assemblers for the Amiga by G. Hull Choose your assembler
Shakeup Replaces Top Management at Commodore by S. Hull
Peter J. Baczor"by S. Hull Manager at CBM gives an inside look
Logistix A review by Richard Knepper
Organize! by A review Richard Knepper database.
68000 Assembly Language Programming on the Amiga
by Chris Martin
Superbase Personal Relational Database by Ray McCabe
AmigaNotes by Rae, Richard A look at FutureSound
Commodore Shows the Amiga 2000 and 500 at the Boston
Computer Society by H Maybeck Tolly

Volume 2, Number 7 1987

New Breed of Video Products by John Foust...
Very Vivid! by Tim Grantham...
Video and Your Amiga by Oran Sands III
Amigas & Weather Forecasting by Brenden Larson
A-Squared and the Live! Video Digitizer by John Foust.
Aegis Animator Scripts and Cel Animation by John Foust
Quality Video from a Quality Computer by Oran Sands III.
Is IFF Really a Standard? by John Foust..
Amazing Stories and the Amiga™ by John Foust.
All about Printer Drivers by Richard Bielak
Intuition Gadgets by Harriet Maybeck Tolley.
Deluxe Video 1.2 by Bob Eller
Pro Video CG1 by Oran Sands III.
Digi-View 2.0 Digitizer/Software by Jennifer M. Janik
Prism HAM Editor from Impulse by Jennifer M. Janik
Easyl drawing tablet by John Foust..
CSA's Turbo-Amiga Tower by Alfred Aburto
68000 Assembly Language by Chris Martin.

Volume 2, Number 8 1987

This month Amazing Computing™ focuses on entertainment packages for the Amiga. Amazing game reviews...

SDI, Earl Weaver Baseball, Portal, The Surgeon, Little Computer People, Sinbad, StarGlider, King's Quest I,II and III, Faery Tale Adventure, Ultima III, Facets of Adventure, Video Vegas and Bard's Tale.

Plus Amazing monthly columns... Amiga Notes, Roomers, Modula-2, 68000 Assembly Language and The Amicus Network.
Disk-2-Disk by Matthew Leeds
The ColorFonts Standard by John Foust
Skinny C Programs by Robert Riemersma, Jr.
Hidden Messages In Your Amiga™ by John Foust
The Consumer Electronics Show and Comdex by J Foust

Volume 2 Number 9 1987

Analyze 2.0 reviewed by Kim Schaffer
Impact Business Graphics review by Chuck Raudonis
Microfiche Filer review by Harv Laser
Pagesetter review by Rick Wirch
Gizmoz Productivity Set 2.0 review by Bob Eller
Kickwork review by Harv Laser
Diga Telecommunications Package review by Steve Hull
Mouse Time and Timesaver review by John Foust
Insider Memory Expansion review by James O'Keane
Microbotics Starboard-2 review by S. Faiwiszewski
Leather Goddesss of Phobos by Harriet Maybeck-Tolly
Lattice C Compiler Version 3.10 reviewed by Gary Sarff
Manx 3.4a Update reviewed by John Foust
AC-BASIC reviewed by Sheldon Leemon
AC-BASIC Compilier an alternative comparison by B Catley
Modula-2 Programming S Faiwiszewski Raw Console Dev. Events
Directory Listings Under AmigaDOS by Dave Haynie
AmigaBASIC Patterns by Brian Catley
Programming with Soundscape Todor Fay manipulate's samples
Bill Volk, Vice-President Aegis Development, by Steve Hull
Jim Goodnow, Devekoper of Manx 'C' interview by Harriet M Tolly
Plus a great collection of monthly columns...

Volume 2 Number 10 1987

Max Headroom and the Amiga by John Foust
Taking the Perfect Screen Shot by Keith Conforti
Amiga Artist: Brian Williams by John Foust
Amiga Forum on Compuserve™... Software Publishing Conference Transcript by Richard Rae
All About Online Conferencing by Richard Rae
dBMAN reviewed by Clifford Kent
Amiga Pascal reviewed by Michael McNeil
AC-BASIC Compiler reviewed by Bryan Catley
68000 Assembly Language by Chris Martin
Amiga Programming:
Amiga BASIC Structures by Steve Michel
Quick and Dirty Bobs by Michael Swinger
Directory Listings Under Amiga-DOS, Part II by Dave Haynie
Fast File I/O with Modula-2 by Steve Faiwiszewski
Window I/O by Read Predmore
Plus a great collection of monthly columns...

Volume 2 Number 11 1987

Word Processors Rundown by Geoff Gamble
ProWrite, Scribble!, and WordPerfect compared
LPD Writer Review by Marion Deland
VizaWrite Review by Harv Laser
Aedit Review by Warren Block
WordPerfect Preview by Harv Laser
Jez San Interview by Ed Bercovitz—StarGlider author speaks!
Do-it-yourself Improvements to the Amiga Genlock
Digi-Paint Review by Harv Laser
Sculpt 3D Review by Steve Pietrowicz
Shadowgate Review by Linda Kaplan
TeleGames Review by Michael T. Cabral
Reason Preview: an intense grammar examination application
As I See It by Eddie Churchill WordPerfect,Gizmoz V2.0 and Zing!
AmigaNotes by R Rae 4 electronic music books
Modula-2 Programming by S.Faiwiszewski devices, I/O, &serial port
68000 Assembly Language by Chris Martin Display routines
The AMICUS Network by John Foust—Desktop Publishing, Seybold
C Animation Part II by Mike Swinger Animation Objects
BASIC Text by Brian Catley Pixel perfect text positioning
Soundscape Part III by Todor Fay VU Meter and more
Fun with Amiga Numbers by Alan Barrett
File Browser by Bryan Catley—Full Feature BASIC File Browsing
Plus a great collection of monthly columns...

Volume 2 Number 12 1987

The Ultimate Video Accessory by Larry White
The Sony Connection by Stewart Cobb
15-Puzzle in AmigaBASIC by Zoltan Szepsi
Life, Part I: The Beginning by Gerald Hull
The ultra-complex nine blit solution to the "Game of Life."
Amiga Virus! by John Foust
CLI Arguments in C by Paul Castonguay
MIDI Interface Adapter by Barry Massoni
Amiga 1000-style MIDI interfaces can fit A2000s or 500s
Modula-2 by S. Faiwiszewski Part 1: command line calculator
AmigaNotes by Rick Rae audio changes made in the A500 &A2000.
Animation for C Rookies: Part III by M. Swinger double-buffering.
The Big Picture by Warren Ring Assembly language programming
Karate Kid Reviewby Stephen R. Pietriowicz
GO! 64 review by John Foust, James O'Keane, and Rick Wirch
Three C-64 experts investigate a new Amiga 64 emulator.
A-Talk-Plus Review by Brendan Larson
Calligrapher Review by John Foust
Animator: Apprentice Review by John Foust
Playing Dynamic Drums on the Amiga by David N. Blank
WordPerfect Review by Steve Hul
Insider/Kwikstart Review by Ernest P. Viveiros Sr
RAM & ROM expansion: Comments and installation tips.
Forth! by Jon Bryan DumpRPort utility for your Multi-Forth toolbox.
As I See It by Eddie Churchill Digi-Paint, Portal, &Videoscape 3D.
The Commodore Show and AmiExpo: New York!
Plus a great collection of monthly columns...

Volume 3 Number 1 1988

AmigaNotes by **Richard Rae** Amiga digital music generation .
C Animation Part IV by Michael Swinger
Forth by John Bryan Sorting out Amiga CHIP and FAST memory
The Big Picture by Warren Ring Daring assembler language programming; CLI system calls and manipulating disk files.
68000 ASsembly Langueage Programming by Chris Martin
"Create a multi-color screen without using Intuition routines!"
Modula-2 Programming by S. Faiwiszewski A new modula-2!
Amicus Network Special Report: Fall COMDEX by J. Foust
The ultimate Video Accessory: Part II by Larry White
Life: Part II by Gerald Hull The Amiga blitter."
FormatMaster: Professional Disk Formatting Engine byC.Mann
Put Batch language to work on the drudgery of disk formatting .
BSpread by Brian Catley full featured AmigaBASIC spreadsheet!
AmigaForum Transcript ed. by Rick Rae Amiga's Dave Haynie.
Halcalc Review by Chuck Raudonis easy to use, spreadsheet.
VIP Professional Review by S. Mitchell Manage stock porfolio
Money Mentor Review by S.Kemp Personal finance system.
Investor's Advantage Review by Richard Knepper
plus "Poor Man's Guide to the Stock Market."
Plus a great collection of monthly columns...

Volume 3 Number 2 1988

Laser Light Shows with the Amiga by Patrick Murphy
Lasers and the Amiga: A Dazzling Tandem
The Ultimate Video Accessory: PartIII by Larry White
Take the final stepstoward designing your own videos.
Our First Desktop Video by Larry White
Step-by -step guide to organizing & presenting your Amiga video.
Hooked on the Amiga with Fred Fish interview by Ed Berkovitz.
Photo Quality Reproduction with the Amiga and Digi-View
by Stephen Lebans
Balancing your Checkbook with WordPerfect Macros by S.Hull
Hand your checkbook worries over to the Amiga.
More Basic Text by Bryan Catley easier text on an Amiga screen
Life: Part III by Gerald Hull
Sries winds up with famed nine-blit calculation & source to LIFER.
Solutions to Linear Algebra through Matrix Computations
by Ropbert Ellis
Simplify matrix algebra with basic operations & routines.
Modula-2 Programming by Steve Faiwiszewski
Catching up with Calc-a source follow-up.
68000 Assembler Language Programming by Chris Martin
Graphics- Part II of Assemgram.
Arazok's Tomb interview by Kenneth E. Schaefer
AIRT by S. Faiwiszewski innovative icon-basede program. lang.
Forms in Flight by S. Pietrowicz Render & Animate 3D objects
Silicon Dreams and the Jewel of Darkness by K E. Schaefer
Leisure suit Larry by Kenneth E. Schaefer
Two New Entries From Microbiotics by John Foust
M501 Expansion & Starboard II MultiFunctionboard.
Mindlight 7 and People Meter by John Foust
Phantasie Ken E. Schaefer Amazing Phantasie Character Editor.
Plus a great collection of monthly columns...

Volume 3 Number 3 1988

Desktop Video,Part IV by Larry White
Put all the pieces together-the desktop video commercial.
The Hidden Power of CLI Batch File Processing by J. Rothman
Make your Amiga easier to use with CLI Batch files.
A Conference with Eric Graham edited by John Foust
The mastermind behind Sculpt 3D and Animate 3D.
Perry Kivolowitz Interviewed by Ed Bercovitz Amiga insights from a major developer and personality.
Jean "Moebius" Giraud Interviewed by Edward L. Fadigan
Avant-garde art comes to the Amiga-in dazzling form.
PAL Help by Perry Kivolowitz A1000 expansion reliability.
Boolean Function Minimization by Steven M. Hart
A useful digital design tool in AmigaBASIC.
Amiga Serial Port and Midi Compatibility for Your A2000! by L. Ritter and G. Rentz Add an A1000-style serial port to the A2000!
Electric Network Solutions the Matrix Way by Robert Ellis
Engineers! Practicle routines for using matrix algebra.
The A.M.U.G. BBS List compiled by Joe Rothman, Chet Solace, & Dorothy Dean 514 BBS phone numbers in the U.S. & Canada.
FACC II reviewed by Graham Kinsey Speed your floppy drives.
Uninvited reviewed by K E. Schaefer
Flow reviewed by Pamela Rothman brainstorms into mental art.
Benchmark Modula-2 Compiler reviewed by Richie Bielak
Modula-2 Programming by Steve Faiwiszewski
The gameport device and simple sprites in action.
AmigaNotes by R. Rae A1000! Software-switchable output filter.
Roomers by The Bandito AmiExpo, Kickstart 1.4, Commodore
The Big Picture by Warren Ring-- Unified Field Theory!!
Plus a great collection of monthly columns...

Volume 3 Number 4 1988

Highlights from AmiExpo, Los Angeles by Steve Hull
Writing a SoundScape Patch Librarian T. Fay System Exclusive
Upgrade Your A1000 to A500/2000 Audio Power—by H.Bassen
Modifications to help your A1000 make sweet music, too!
Amiga Audio Guide Listing of all Amiga audio products.
Gels in Multi-Forth by John Bushakra
Macrobatics by Patrick J. Horgan Ease the trauma of assembly language programming.
Amiga Audio Sources The folks behind all those audio products.
Take Five! by Steve Hull five Amiga games reviewed.
Amiga Notes by Rick Rae A basic tour of Amiga audio.
The Ultimate Video Accesory, Part V by Larry White
Bug Bytes by John Steiner
The Big Picture by Warren Ring Part II Unified Field Theory.
Roomers by The Bandito
Hardware hijinx ... Toasted video ... the dream Amiga ... and more!
In the Public Domain by C.W. Flatte
Time Bandit review by Keith Conforti
AudioMaster review by B. Larson Real-time digitizing samples.
Music Mouse review by J Henry Lowengard
Making music without lifting a finger from the mouse.
Amiga-Tax/Canadian Version review by Ed Bercovitz
A Canadian income tax planning, preparation, & analysis package.
SAM BASIC review by Bryan Catley
A new BASIC which exploits even more unique Amiga features.

Volume 3 Number 5 1988

Interactive Startup Sequence by Udo Pernisz
The Command Line part 1by Rich Falconburg
AmigaTrix III by Warren Block—Tips and tidbits to ease Amiga life
Amiga Product Guide: Hardware Edition
Proletariat Programming by P Quaid—Public domain compilers
The Companion by P.Gosselin Amiga's Event Handling capability.
MindLight 7 reviewed by David N. Blank
VideoScape 3-D 2.0 reviewed by David Hopkins
Extend reviewed by Bryan D. Catley—An AmigaBASIC extension
AssemPro reviewed by Stephen Kemp
Opening a door to assembly language programming.
APL.68000 reviewed by Roger Nelson
Book Reviews by Richard Grace—Three "C" programming texts.
CBTREE reviewed by Michael Listman
A tidy collection of functions to aid the C programmer.
The Big Picture by Warren Ring
The three-part Unified Field Theory winds up
Modula-2 by Steve Faiwiszewski
Termination modules for Benchmark and TDI compilers.
68000 Assembly Language by Chris Martin
Peeling away the complication of display routines.
Plus a great collection of monthly columns...

Volume 3 Number 6 1988

Bear Time Reviewed by Steve Carter
What makes this inexpensive A1000 battery-backed clock tick?
Acquisition Reviewed by David N. Blank
A look inside the latest release of a powerful relational database.
Butcher 2.0 Reviewed by Gerald Hull
A tidy collection of diverse image processing utilities.
Reassigning Workbench Disks by John Kennan
Endless disk swapping comes to a merciful end.
Product Guide: Software Tools Edition
A listing of all the products you need to put your Amiga to work.
An IFF Reader in Multi-Forth by Warren Block
Create an easy to use IFF reader in Multi-Forth.
Basic Directory Service Program by Bryan Catley
A programming alternative to the GimmeeZeroZero windows.
C Notes from the C Group by Stephen Kemp
A beginner's guide to the power of C programming.
An Amiga Forum Conference with Jim Mackraz
The Amiga market as seen by the "Stepfather of Intuition."
Son of Seven Assemblers Reviewed by Gerald Hull
A ocmparative battle between seven native-code assemblers.
The 1988 Commodore Amiga Developers Conference
A look inside the conferences held in Washington, D.C.
Amiga Working Groups by Perry Kivolowitz and Eric Lavitsky
An outline of the innovative Amiga Working Groups concept.
The Command Line by Rich Falconburg
Exploring the multi-talented LIST command.
Plus a great collection of monthly columns...

Volume 3 Number 7 1988

Look, Up On the Screen, It's an Ami... It's a Pro... It's SuperGen reviewed by Larry White—Genlock comparisons
An Interview with "Anim Man," Gary Bonham by B. Larson
An animated conversation with the man behind the format.
The Amiga at Spring COMDEX in Atlanta by Ed Bercovitz
Amiga Product Guide: Video/Graphics Edition
Thirteen pages devoted to the Amiga's dazzling strong suit.
The Developing Amiga by Steve Pietrowicz
Developers' notes: PD vs. shareware vs. freely distributableetc.
Roll Those Presses! by Barney Schwartz
Welcome to the dandy, demanding world of desktop publishing!
Linked Lists in C by W. E. Gammill Put dynamic memory to work!
FrameGrabber Preview by Oran Sands
Capturing an image can now be as fast as punching a single key!
A First Look at Interchange reviewed by David Hopkins
Bridge the gap between those incompatible animation packages.
Perfect Vision reviewed by Bryan Catley
Capture, digitize and save pictures from any video source.
ProWrite 2.0 Review reviewed by Pamela Rothman
A graphic word processor specializing in efficient editing.
Doug's Math Aquarium: The Art of Mathematics by R. Bielak
Bear Products MegaRex II Expansion RAM by Steve Carte
The Command Line by Rich Falconburg
Amiga Notes by Rick Rae
The Other Guys' Synthia: A digital synthesizer features won't stop.
C Notes from the C Group by Stephen Kemp
Weathering the unknown "C" of basic object and data types.
Plus a great collection of monthly columns...

Volume 3 Number 8 1988

The Command Line by Rich Falconburg
The journey into the CLI continues.
The Developing Amiga by Stephen R. Pietrowicz
A gaggle of great programming tools.
Modula-2 Programming by Steve Faiwiszewski
Libraries and the FFP and IEE Math Routines.
C Notes from the C Group by Stephen Kemp
Arrays and pointers unmasked.
Dark Castle reviewed by Keith Conforti—The Black Knight lurks
Ports of Call reviewed by Julie Landry
Leatherneck reviewed by Michael Creeden-Rambo's not so tough!
Capone reviewed by Joyce and Robby Hicks- Light Guns blaze
Casino Fever reviewed by Michael T. Cabral—Vegas on Amiga
Ferrari reviewed by Jeffery Scott Hall—Start your engine
Arkanoid reviewed by Graham Kinsey—"blockbuster"
Ebonstar by Keith Conforti —black hole treking.
Deluxe Productionsreviewed by Harv Laser—Video wizardry
Game Pizazz by Jeffery Scott Hall— Register your questions here.
TrackMouse by Darryl Joyce
Convert a standard Atari trackball into a peppy Amiga TrackMouse.
Amiga Interface for Blind Users reviewed by Carl W. Mann
An ingenious interface that opens the Amiga to even more users!
Video in the Sunshine State reviewed by Stephen R. Pietrowicz
RGB Video Creations hosts a video unveiling!
Amiga Product Guide: Games Edition
Tumblin' Tots by David Ashley—assemply language program.
Plus a great collection of monthly columns...

Volume 3 Number 9 1988

The Kideo Tapes by John Dandurand
A Georgia elementary school puts desktop video to work.
Speeding Up Your System by Tony Preston floppy disk caching
Amiga Product Guide: Education Edition
Everything you need to send your Amiga to the head of the class.
Computer Aided Instruction by Paul Castonguay
Authoring system in AmigaBASIC.
Gels in Multi-Forth, Part II: Screenplay by John Bushakra
Make the IFF converter from Part I easy to use-gadgets, menus,etc.
AmiExpo Midwest '88 by Michael T. Cabral
After taking the coasts by storm, the Amiga wows Chicago
Intellitype by Harv Laser—Learning to type made easy...and *fun?*
Shakespeare by Barney Schwartz—Desktop publishing in full color.
XSpecs 3D by Steve Hull—A new dimension in Amiga graphics.
AmigaNotes by Richard Rae—How IFF sound samples are stored?
Take Five! by Steve Hull—Beat the back-to-school blues !
The Command Line by Rich Falconburg—continuing tour of CLI.
Hot on the Shelves by Michael T. Cabral & Michael Creeden
What do you get when you combine intense war strategy with a monochrome monitor and desktop presentation? Check it out.
Bug Bytes by John Steiner
C Notes from the C Group by Stephen Kemp
Operators, expressions, and statements in C uncovered.
Roomers by The Bandito Can Apple IIgs Plus keep Amiga away?

Volume 3 Number 10 1988

A First Look At Deluxe PhotoLab reviewed by David Duberman
A paint package, poster-maker, and image processing program
DiskMaster reviewed by Steve Hull —file management utility.
DSM: A MC680000 Disassembler reviewed by Gerald Hull
Looking for easily modifiable, assembler-ready code?
FBasic Language System reviewed by Patrick Quaid
BASIC compiler and development system.
Hot on the Shelves by Michael T. Cabral—Deviant dice, gripping gray scales, color cartography, mauling modems, and much more.
The Command Line by Rich Falconburg
NEWCLI: A painless way to create a new console window.
The Developing Amiga by S. Pietrowicz Usenet—24-Hour News
C Notes from the C Group by Stephen Kemp —loops
Roomers by The Bandito WP wars, ignominious interfaces, & more
PD Serendipity by C.W. Flatte--Fred Fish collectionpasses 150.
Comparison of MultiScan Monitors by Steven Bender
Five multiscan alternatives square off on the desktop.
Record Keeping for Free-lancers:
A Superbase Professional Tutorial by Marion Deland
Record keeping system for free-lance photographers and others.
On The Crafting of Programs by David J. Hankins—A look at optimization kicks off a series of articles on programming savvy.
Bob and Ray Meet Frankenstein by Robert D'Asto—Create, animate, and metamorphose graphics objects in AmigaBASIC.
Digital Signal Processing in AmigaBASIC by Robert Ellis
Perform your own digital experiments with Fast Fourier Transforms.
HAM & AmigaBASIC by Bryan Catley—Pack your AmigaBASIC programs with many of the Amiga's 4096 shades!
CAI—Computer Aided Instruction: Part II by Paul Castonguay
The Editor program wraps up our authoring system in AmigaBASIC.

Volume 3 Number 11 1988

Desktop Publishing with Professional Page by Barney Schwartz tutorial in document creation, plus some jazzy enhancements.
Game Pizzazz by J. Hall gaming hints, tips, high-score secrets.
Structures in C by Paul Castonguay C programming in an nutshell.
On The Crafting of Programs by D. Hankins speed up your progs.
Desktop Video VI: Adding the Third Dimension by Larry White
Unraveling the complexity of 3D for your video creations.
A2000 Hard Drive Round Up by Sheldon Leemon
Keyclick by Mike M. Duppong a typewriter click in your keyboard.
More Linked Lists in C: Techniques and Applications by Forest W. Arnold Procedures for managing lists, storing diverse data types in the same list, and putting lists to work in your programs.
BASIC Linker by Brian Zupke Combine individual routines from your program library to create an executable program.
The Developing Amiga by Steven Pietrowicz
A look at mysteries and successes behind efficient beta testing.
Modeler 3D Preview reviewed by David Hopkins
A peek inside a new, open-ended 3D package.
AProDraw Graphics Tablet reviewed by Keith Conforti
Artists! Meet the future of Amiga graphics.
StarGlider II reviewed by Jeffery Scott Hall
Those irritating Ergons are back for another laser-lashing.
WShell reviewed by Lawrence Lichtman CLI substitute,
Hot on the Shelves by M. Cabral viruses, music, microfiche mastery
PD Serendipity by C.W. Flatte Fred Fish disks 149-152.
Roomers by The Bandito Golden RAM, 16-bit videogames, CD-I, another HAM skirmish ... what could possibly be NeXT?

To be continued.........

To Order
Back Issues,
please use the order
form on page 112

(*Menus*, *continued from page 79*)

number in this location when the user selects more than one menu selection at once by clicking the left mouse button on them.

This may all seem complex at first but it gets much simpler (honest) after you've had a little practice using this information and seeing the results. The "big picture" regarding these structures is analogous to the entire Menu display itself. Each Menu Structure is linked to (contains the memory address of) the Menu Structure to it's right. Each Menu Structure is also linked to the structure of the first MenuItem beneath it. Each MenuItem structure within a particular Menu is linked in a "vertical" list and, if a MenuItem has a list of SubItems, these are also in a linked list and the MenuItem structure contains the address of the first one of these SubItems. This entire network of linked structures describes the whole menu display and is known as the MenuStrip. The beginning of this MenuStrip, and thus its memory address, is the first field of the first Menu Structure within it.

More information can be gotten on all these structures and fields as well as other detailed aspects of Intuition from the Intuition Reference Manual by Robert J. Mical and Susan Deyl, published by AddisoncWesley, a fine text and definitely a must read for all Amiga programmers. The purpose of this article is to give the AmigaBASIC programmer a general grasp of the principles involved with menu creation and, hopefully, to encourage further study and experimentation.

There's one other Intuition structure intimately involved with creating menus: The IntuiText Structure. The fourth field of the Menu Structure and the fifth field of the MenuItem Structure point, not simply to a text string, but to an IntuiText Structure which contains data describing how the text is displayed: Colors, text style, and so on. The specifics on this structure are as follows:

IntuiText Structure:
field #1—two 1-byte numbers: fgColor, bgColor
field #2—a 1-byte number stating the Draw Mode of the text
field #3—2-byte number: LeftEdge
field #4—2-byte number: TopEdge
field #5—4-byte number: pointer to a font structure
field #6—4-byte number: pointer to text string
filed #7—4-byte number: pointer to next IntuiText struct

The first field is just the PALETTE-id numbers as normally used in AmigaBASIC for the desired foreground and background colors of the text. Field #2 is the Draw Mode. AmigaBASIC uses Draw Mode 1 as a default. Draw Mode 0 is usually better for menus as it doesn't automatically place a background color behind the printed text. Fields #3 and #4 are text placement parameters. The fifth field is beyond the scope of this article, but you don't have to worry about putting any value here as a "0" will produce the default topaz font. Field #6 is the memory address of the null-terminated text string, which means "textstring$ + CHR$(0)" in AmigaBASIC. The last field is reserved for the address of another IntuiText Structure in case you want a linked list of these structures, which isn't needed for menus.

(*continued*)

Another **NEW** Product
To Make The **FASTEST** Development
Environment Even **FASTER**!

The **F-BASIC**™ Source Level DeBugger (SLDB)

The Only DeBugger Quick Enough For The **F-BASIC**™ System

Finally, a full featured DeBugger that doesn't require a working knowledge of assembly code. Use the SLDB to quickly debug your F-Basic programs without all those PRINT statements. Check these features of SLDB:

- Debug F-Basic programs at the SOURCE level.
- Fully windowed Intuition interface—windows for Source Code, 68000 Registers, Memory Dumps, Program Variables, etc.
- Set break points and single step trace by either source line or by Assembly Code Line.
- Display all variables, arrays, or RECORDS by name.
- Full reverse-assembler included.
- A comprehensive User's Manual with full documentation of all SLDB supported features and windows.

The **F-BASIC**™ Language System

An Enhanced Compiled BASIC Language
You've **READ** About It—Now **EXPERIENCE** It!

☐ Beginning AMIGA™ programmers choose F-Basic due to its simplicity and ease of use. If you know BASIC, you can use F-Basic immediately.

☐ Programmers of all levels choose F-Basic because its hundreds of built-in features provide an AMIGA language environment that coincides with your experience. F-Basic can't be outgrown, due to features like:

- Extensive Control Structures
- LOCAL & GLOBAL Variables
- Ultra Fast Floating Point
- RECORD Structures & Pointers
- INCLUDE & APPEND Separate Files
- PATTERN Matching Support
- Recursive SUBROUTINES & FUNCTIONS
- 32, 16, and 8 Bit INTEGERS
- Powerful String Facilities
- Direct ROM Kernel Access
- Bitwise Operations
- Access to 68000 Registers
- Easy high level access to AMIGA Screens, Windows, Menus, Sound, Speech, Graphics and Events.

☐ The F-Basic™ Language System includes:
  - Comprehensive User's Manual—Full Documentation of All Supported Features Of The F-Basic System
  - Sample Programs Disk—"Hands On" Tutorial
  - FastCom™ single pass compiler, which quickly compiles F-Basic programs into assembly code. FastCom is written in F-Basic... That's your guarantee of the speed and efficiency of F-Basic.

The **F-BASIC**™ Language System—Only $79.95
F-BASIC™ System and Complete SLDB—Only $129.95

SEND CHECK OR MONEY ORDER TO:

DELPHI NOETIC SYSTEMS, INC.

Post Office Box 7722
Rapid City, South Dakota 57709-7722
Credit Card or C.O.D. Call (605) 348-0791

F-BASIC and FastCom are registered trademarks of DNS, Inc.
AMIGA is a registered trademark of Commodore/AMIGA Inc.

Okay, that's the lowdown on the Intuition structures directly related to menus. There are other system structures that come into play but it isn't necessary to have a complete listing of them here. There is, in fact, an entire complex of interlinked structures created automatically every time we boot up the Amiga which defines its basic operations. Every screen, window, gadget and icon has a structure which defines it; and there are many other structures used for graphics display, inter-system communication and many other activities. These structures are defined in the above Intuition manual as well as the ROM Kernel Manual and other texts. They are listed in these volumes in the chapters labeled "Include Files" and are written in C language, but it isn't necessary to learn C to be able to use them. All that's needed is some familiarity with the C terms used to denote variable types and pointers so that the fields of these structures can be understood as to size and type. Any basic C text will have definitions for these terms. From there it's just a matter of locating the addresses of the structures one wishes to manipulate.

The AmigaBASIC programmer has three ways of getting this address information. The functions WINDOW(7) and WINDOW(8) return the addresses of two often-used structures: The Window Structure mentioned earlier (for defining Windows) and the RastPort Structure (defines graphic drawing information). The second method is to use one of several System Routines which were created for the purpose of providing structure addresses for the programmer (more on this in a moment). The third method of locating a structure address is to locate a field within one structure which is known to contain a pointer to the structure you seek. I'll give you an apropos example of this last method. Let's say we wanted to find the address of an existing MenuStrip. We know that AmigaBASIC's WINDOW(7) function returns the address of the Window Structure. It so happens that the Window Structure contains the address of the MenuStrip within a 4-byte field that begins 28 bytes from its "top". Thus, the address of the MenuStrip can be gotten with this code fragment:

```
PRINT PEEKL(WINDOW(7)+28)
```

Here we're using one structure to locate the address of another.

You might be wondering why we can't just look up the address of a structure in some manual when we need to access it for programming purposes. This can be done with other computers, why not the Amiga? The reason is, because of its multi-tasking nature, the Amiga does not use absolute (fixed) addresses in its system. It uses relative addressing instead. This means that memory addresses are established in relation to some other address. An analogous example: How do you find the end of a yardstick? Easy. Just find the beginning and measure 36 inches.

The MenuNumber and Other Oddities

If you're with me so far you may still be wondering how we can set up a program so that it can detect when a SubItem has been chosen by the user. AmigaBASIC commands only include the MENU(0) and MENU(1) functions for returning Menu and Item choices. Is there some secret BASIC function for detecting SubItems as well? No, there isn't. Fortunately, though, Intuition is kind enough to keep track of this information for us and to make it available to programs which know where to find it and how to interpret it. I use the word "interpret" because the information itself exists in a sort of codified form and must be converted before being used by the program. First I'll explain the form of the information and then I'll cover where it can be located.

When we're dealing with menu selections we have three possible variables: Menu, Item and SubItem. These values comprise relatively small numbers and could be recorded using single byte numbers for each, a total of three bytes, to give a complete description of a menu selection made by the user. Intuition, though, doesn't like the idea of using three bytes for this data; it prefers to use only two bytes. So, it "packs" these three numbers into a single word (two-byte area) of memory and the result is one 16-bit binary number which describes the three values we seek. This "packed number" is referred to as the MenuNumber and here is a binary example of what one might look like (the spaces are for clarity only):

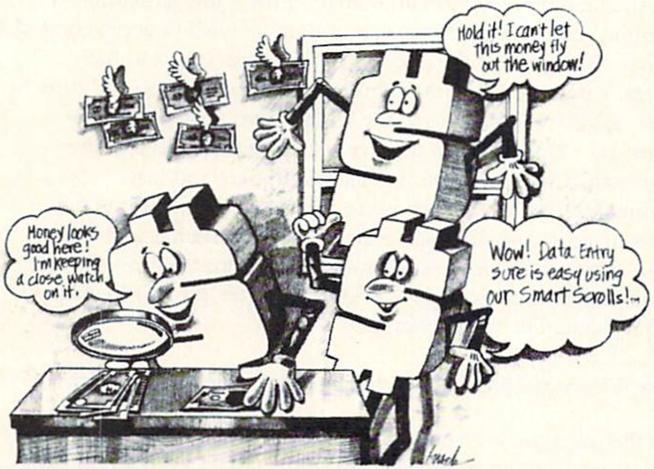
```
0000 1000 0100 0011
```

At first glance it doesn't look much like three different numbers...four, perhaps, or maybe two...but three? Yes, it is three different numbers. This MenuNumber tells us that the user has chosen Menu 3, Item 2, SubItem 1. How? Let's chop this number up a little and it will become apparent.

```
0000 1   000 010   0 0011
```

We've broken up the 16-bit number into three parts. The middle part has 6 bits, the end parts each have 5 bits. The group of five bits to the right gives the binary number of the Menu, the middle group of six bits is the Item, and the five bits on the left represent the SubItem. Taken all at once this example of a MenuNumber is equal to 2115 in decimal or &H0843 in hex, but this value is not terribly important to us (yet); right now we just want the three individual numbers so our program will know where to jump when the user makes a selection. An important point to remember is that these individual numbers begin counting at zero. The first Menu, Item or SubItem is always numbered as zero, number 1 will be the second and so on. This, then, leaves us with the need of a way to represent "no choice" as something other than zero. For example, if the user picks an Item which has no SubItems we don't want the SubItem part of the MenuNumber to register a zero as that would mean that the first SubItem was selected, which would make no sense as there are no SubItems in this case. To take care of this situation Intuition would fill that part

Meet a team of the friendliest financial organizers you'll ever run across.



Hold it! I can't let this money fly out the window!
Money looks good here! I'm keeping a close watch on it.
Wow! Data Entry sure is easy using our Smart Scrolls!

When you want to manage your personal finances, Money Mentor goes a step beyond.

Plug Money Mentor into your Amiga and a virtual teamwork effort takes place in watching over every aspect of your personal finances.

The new "C" version of Money Mentor is the friendliest financial organizer obtainable today!

Now you can experience super-speed data entry, dazzling graphic output and an extremely friendly attitude!

Smart Scrolls for speed.

Money Mentor has a truly unique system called *Smart Scrolls*, that handles a diversity of otherwise tedious data entry functions and clips along saving you up to 70% of your typing time. It's a *smart* addition to Money Mentor, that's why we call it *Smart Scrolls*.

Money Mentor Features:

- Net Worth Statement
- 200 Budget Categories
- 30 Integrated Accounts such as Checking, Cash, Savings and Credit Cards
- Elaborate Search Routine allows editing of transactions according to your specific guidelines
- Automatic Check Printing
- Automatic Account Balancing
- Color Graphic Reports illustrating *actual vs. budgeted* amounts
- Over 50 Reports to choose from!

What they're saying about us!

"Money Mentor has to be the nicest look and feel of any money manager package for home use that I have ever seen." — Amiga Sentry

"Money Mentor is an excellent product" — Amazing Computing

Money Mentor is for everyone!

It does more than just keep your checkbook balanced. Money Mentor helps you manage your personal finances which is important to any family or individual.

With Money Mentor, you can be looking better financially.

Order Money Mentor today.

Money Mentor sells for only $95.95!

VISA MasterCard

SEDONA SOFTWARE

SEDONA SOFTWARE/11828 RANCHO BERNARDO RD., SUITE 128-20/SAN DIEGO, CA 92128/CALL (619) 451-0151

of the MenuNumber with ones to represent a "null" choice. If nothing at all was chosen (right mouse button pushed and released with nothing selected) the entire MenuNumber would be set as ones. Here's an example of a MenuNumber which contains a null SubItem:

```
1111 1   000 001  0 0000
```

This translates to Menu 0 (left most Menu), Item 1 (second Item) and SubItem null (no SubItems involved).

Our program code contains three subprograms which extract these three values from the MenuNumber for us: MENUNUM, ITEMNUM and SUBNUM. These represent AmigaBASIC "translations" of the functions a C programmer uses to obtain these same values. The only one a BASIC programmer really needs is SUBNUM, since the MENU(0) and MENU(1) functions provide Menu and Item choices. All three are given here for the sake of completeness and to assist in further explorations by the reader.

Now that you know how to interpret MenuNumbers all that's needed is the location of this MenuNumber so it can then be read and used by the program. This information can be found within a special system structure known as the MessageKey Structure. It's not necessary to have all the details of this structure here. All we need to know is where it is and where, within it, the MenuNumber can be found. The address of this MessageKey structure resides within the Window Structure mentioned above, to be exact, in the field beginning at "Window Structure + 94". Additionally, the precise information we want is contained in the field which is 24 bytes from the beginning of the MessageKey Structure. The MenuNumber, then, can be found with this coding:

```
MessageKey&=PEEKL(WINDOW(7)+94)

MenuNumber%=PEEKW(MessageKey&+24)
```

Notice the "&" and "%" endings to the above variables and the two variations of the PEEK function used. Addresses are long integers and the MenuNumber is a short (16 bit) integer. After the user has made a Menu selection we simply use the above code to obtain the MenuNumber and pass it on to the MENUNUM, ITEMNUM and SUBNUM subs to extract the appropriate values.

(continued)

All that's needed now is a description of the library routines which are used in the accompanying code and you're well on your way to being a menu artiste extrordinaire. These ROM Routines are simply subprograms and are used in much the same way as AmigaBASIC subprograms. There are two general types: The type that performs a task and the type that returns a value (some do both). The first type always has an "=" in their syntax and they must be declared with a DECLARE FUNCTION statement before using them. This simply declares whether the value returned will be a short or long integer by tagging on a "%" or a "&" at the end of its name in the declaration. The other type is simply called with the same syntax as AmigaBASIC subprograms and requires no declarations.

I'll give a brief description of each of the system routines used in our code example.

AllocMem is an exec function and it, as stated above, allocates a block of memory for a specific use and returns the address of that block. It needs two things to do its job: The length (in bytes) of the desired block and a number which corresponds to one or more options. Its syntax is:

```
Address&=AllocMem&(length,option&)
```

The value denoting the length should be in multiples of 8 bytes, if not it will automatically be rounded up to the next multiple. The option parameter determines the general location of the desired memory and its status. For our purposes we use the option with the value of 65537 (or 2^16 + 2^0) which tells AllocMem to clear the allocated memory to zero and keep it from being shifted around. This gives us a "clean slate" to create our structure and a permanent address with which to access it. There are other possible option values but they don't concern us here. The "Address&" variable (or any other we choose) will be automatically assigned the value corresponding to the address of the memory block provided by the function.

Another very useful exec routine is called FreeMem. This one isn't used in our SubMenu_Demo listing, but it should be known if one is starting to work with structures. It requires no DECLARE FUNCTION statement and its syntax is simply:

```
FreeMem address&, byteSize
```

where "address&" is the starting address of the block of memory one wishes to clear and "byteSize" is its length in bytes.

It's always good programming practice to free up any RAM previously allocated before ending or exiting a program. The Amiga system provides a variety of ways to both allocate and clear memory. Our SubMenu_Demo listing doesn't use FreeMem as there is a special routine (described below) for clearing menu structures.

One other exec memory handler which is used in our lising is CopyMem. This routine copies one area of memory into another. It's syntax is:

```
CopyMem& source&, destination&,size
```

where "source&" is the address of the block you wish to copy and "destination&" is where you want the copy to reside. "Size" is the length of the block in bytes. It's very useful when you need several copies of a particular structure.

ItemAddress is an Intuition function which provides the memory address of any MenuItem or SubItem structure. Since a complete menu display can contain quite a few structures, this routine can be a real time saver when we wish to access one of them. We "feed" it the address of the MenuStrip and the MenuNumber of the desired item and it produces the address of that item by assigning it to the variable we provide for that purpose. This is where the total value of the MenuNumber comes in and the reason we need to understand how MenuNumbers work in order to use this routine. When the structure address of a particular Item or SubItem is needed we must figure out what the MenuNumber of that item would be, compute its total value and insert this value into the ItemAddress statement. Its syntax is:

```
MenuItemAddress&=ItemAddress&(MenuStrip&,MenuNumber)
```

In the parentheses we provide the address of the MenuStrip and the MenuNumber of the desired item. The variable on the left will then be assigned the value of the memory address of that item's structure.

The Intuition routine ClearMenuStrip does just what it sounds like: It clears the entire MenuStrip from memory in one fell swoop. It needs only the address of the Window Structure to do its job, so its syntax is simply:

```
ClearMenuStrip& WINDOW(7)
```

since "WINDOW(7)" provides the Window Structure address for us. Notice that we can mix AmigaBASIC and system functions in the same statement. Use ClearMenuStrip in the exit routine of your program or before replacing one menu display with another.

Now, let's scan though the various routines and subprograms of our code. The heart of this system consists of the subprograms SUBMENU and SubMenuSet. The first is used for each SubItem we wish to include in a particular SubMenu. As such it is used in a manner similar to the standard MENU statements. Be sure to number your SubItems correctly and consecutively when using it. The syntax is:

```
SUBMENU SubMenuNum%, SubMenuText$
```

The first parameter is the number of the SubItem in a SubMenu list (start counting at 0) and the second is the text of the SubItem title which should be enclosed in quotes. That's all there is to it. Right after the last SUBMENU call of a given list of SubItems we use SubMenuSet to link all the SubItems just given into a linked list and to link this entire list to the desired MenuItem. The syntax is:

```
SubMenuSet MenuNum%, ItemNum%
```

The parameters are simply the numbers that correspond to the Menu and Item to which we wish to add the list of SubItems. Remember, the first Menu, Item or SubItem is always considered to be "0". This is important because AmigaBASIC starts numbering its Menus at "1", yet it starts its Items with "0", an annoying inconsistency. If you have trouble remembering that AmigaBASIC MENU 1 is really MENU 0 and prefer to keep the numbering system consistent with AmigaBASIC's you can do so by adding a line of code at the beginning of the SubMenuSet subprogram which subtracts 1 from the Menu number parameter.

Tweaks and Tricks

The POKEW statements within the SUBMENU sub are labeled as to which parameter they control. The first one (Left Edge) is the number of pixels between the left edge of the Menu and the left edge of the SubMenu. This is arbitrarily set at "68" in the code listing because that's what works with this particular demo. Your program may well need another value here, so change it to suit your program's needs. The three following POKEW statements are various dimension parameters of the box that holds the SubItem title, also measured in pixels. Try different values and you'll see how they work. The "Flags" parameter describes various display options. Space does not permit a complete rundown on all the possible values and combinations that can be used here, besides...I'd be spoiling the fun by telling you everything! The Intuition manual has all the data about the various Flags values in its "Include Files" section. I will give you one more little tidbit, though: try erasing the "&H40" and substituting "&H80" in its place. Now, run the program and move the mouse pointer over the Menu selections. Do you see the difference in the way the items are highlighted?

The CreateIntuiText sub, also containing some noteworthy parameters, is not called directly by the program, but rather by the SUBMENU sub. It does exactly what its name implies: It creates an IntuiText structure for the text of each SubItem and then passes the address of that structure back to the SUBMENU sub so it can then "plug" this address into the appropriate field of the SubItem structure. Most of its parameters are set to "0" in our example code and, as such, don't even need to be there. I've included them to give you the option of trying different values in these positions.

For example, the first two (Front Pen and Back Pen) are just PALETTE-id numbers. These values will depend on how the SCREEN depth and PALETTE statements are set up in your particular program. You want a chartreuse-on-turquoise menu? Go for it!

Draw Modes are different methods of rendering text or image data on the screen. The possible values are 0, 1, 2, 3 and combinations (sums) of these values. The modes these values represent are called JAM1, JAM2, COMPLEMENT and INVERSVID respectively. Again, space does not permit a complete explanation of these modes here. JAM1 works well with menus but feel free to experiment.

The "taPtr&" means "text attribute pointer" and is used to incorporate different text styles into the Menu. You're on your own on this one. Try the ROM Kernel Reference Manual and/or the Amiga Programmer's Handbook. A pot of coffee may help. In the mean time you can just keep this parameter set at "0" for good ol' topaz.

3D OPTIONS

CONVERT IFF BITMAPPED PICTURES TO
2D AND 3D OBJECTS!!!

NOW YOU CAN MAKE USE OF ALL THOSE
DPAINT AND DIGITIZED PICTURES IN
YOUR FAVORITE 3D OR CAD PACKAGE.

VIDEOSCAPE 3D, AEGIS DRAW, INTROCAD, DCAD
PROFESSIONAL PAGE, POSTSCRIPT
DXF (AUTOCAD...) & MORE.

TO ORDER CALL
1 800-628-2828
EXT 829
WE ACCEPT
VISA, MASTER CARD AND AMERICAN EXPRESS

ONLY $49.95

OR SEND CHECK OR MONEY ORDER TO:

RAINBOWS EDGE PRODUCTIONS
4412 4TH AVE SUITE 2
BROOKLYN N.Y. 11220
Also available: 3D MAPS, 3D FONTS, 3D UTILITIES ($29.95)
All product names are Trademarks of their respective companies.

As stated earlier, the MENUNUM, ITEMNUM and SUBNUM subs are there to extract the appopriate values corresponding to the Menu, Item and SubItem so your program knows where to jump after a selection has been made by the user. This is done by performing the necessary bit manipulation of the MenuNumber value and assigning the result to the variables MenuPick, ItemPick and SubPick. Note that a null SubItem selection will equal 31 (all 5 SubItem bits set to one), so a trap is included in the code to provide an avenue of execution for this possibility.

This information is by no means all there is to know about menus, structures or Intuition. It's meant to be an introduction to these subjects only and a spark for further study. For those AmigaBASIC programmers interested in expanding their grasp of the operating system, Intuition is a good place to start. Its objects are familiar and its routines are relatively easy to learn.

The "missing link" for many is the structure. Learn it well and you will dispel much of the Amiga system's mystery.

(continued)

Listing One

```
'***  SubMenu_Demo  ***

'***  source code   ***
'***       by       ***
'***  Robert D'Asto ***

DECLARE FUNCTION AllocMem&() LIBRARY
DECLARE FUNCTION ItemAddress&() LIBRARY

LIBRARY "exec.library"
LIBRARY "intuition.library"


SCREEN 1,640,200,2,2
WINDOW 2,"SubMenu Demo",,1,1

ON BREAK GOSUB EndIt
BREAK ON

PRINT "Select a Menu item"
PRINT "To quit press CTRL-C"
PRINT

MENU 1,0,1, "  Menu 0 "
MENU 1,1,1, "  Item 0 "
MENU 1,2,1, "  Item 1 "
MENU 1,3,1, "  Item 2 "
MENU 1,4,1, "  Item 3 "
MENU 1,5,1, "  Item 4 "
MENU 1,6,1, "  Item 5 "

SUBMENU 0,"subItem 0"
SUBMENU 1,"subItem 1"
SUBMENU 2,"subItem 2"
SubMenuSet 0,0

SUBMENU 0,"subItem 0"
SUBMENU 1,"subItem 1"
SUBMENU 2,"subItem 2"
SubMenuSet 0,2

SUBMENU 0,"subItem 0"
SUBMENU 1,"subItem 1"
SUBMENU 2,"subItem 2"
SUBMENU 3,"subItem 3"
SubMenuSet 0,5

MENU 2,0,1, "  Menu 1 "
MENU 2,1,1, "  Item 0 "
MENU 2,2,1, "  Item 1 "
MENU 2,3,1, "  Item 2 "
MENU 2,4,1, "  Item 3 "
MENU 2,5,1, "  Item 4 "

SUBMENU 0,"subItem 0"
SUBMENU 1,"subItem 1"
SUBMENU 2,"subItem 2"
SubMenuSet 1,1

SUBMENU 0,"subItem 0"
SUBMENU 1,"subItem 1"
SUBMENU 2,"subItem 2"
SUBMENU 3,"subItem 3"
SubMenuSet 1,3

MENU 3,0,1, "       "
MENU 4,0,1, "       "

ON MENU GOSUB MenuSort
MENU ON

loop:GOTO loop

MenuSort:
MessageKey&=PEEKL(WINDOW(7)+94)
MenuNumber%=PEEKW(MessageKey&+24)
MENUNUM MenuNumber%
ITEMNUM MenuNumber%
SUBNUM MenuNumber%
PRINT "You picked Menu:";MenuPick
PRINT "             Item:";ItemPick
IF SubPick <>31 THEN
  PRINT "             SubItem:";SubPick
ELSE
  PRINT "             SubItem: none"
END IF
RETURN

SUB SUBMENU (SubItemNum%,subTitle$) STATIC
IF SubItemNum% > 9 OR SubItemNum% < 0 THEN EXIT SUB
IF SubItemNum%=0 THEN DIM SHARED AddrArray&(9)
opt&=2^0+2^16
AddrArray&(SubItemNum%)=AllocMem&(32,opt&)
MI&=AddrArray&(SubItemNum%)
POKEW MI&+4,68                'Left Edge
POKEW MI&+6,SubItemNum% *10   'Top Edge
POKEW MI&+8,LEN(subTitle$)*9  'Width
POKEW MI&+10,8                'Height
POKEW MI&+12,&H2+&H10+&H40    'Flags
CreateIntuiText subTitle$,ITaddr&
POKEL MI&+18,ITaddr&
END SUB

SUB SubMenuSet (MENUNUM%,ITEMNUM%) STATIC
MenuNumber%=&HF800+(32*ITEMNUM%)+MENUNUM%
ms&=PEEKL(WINDOW(7)+28)
IA&=ItemAddress&(ms&,MenuNumber%)
POKEL IA&+28,AddrArray&(0)
n=0
WHILE AddrArray&(n)<>0 AND n<10
POKEL AddrArray&(n),AddrArray&(n+1)
n=n+1
WEND
ERASE AddrArray&
END SUB


SUB CreateIntuiText (Text$,ITTA&) STATIC
Text$=Text$+CHR$(0)
Size&=LEN(Text$)+5
opt&=2^0+2^16
IT&=AllocMem&(24,opt&)
TextAddr&=AllocMem&(Size&,opt&)
CALL CopyMem&(SADD(Text$),TextAddr&,Size&)
POKE IT&,0    'Front Pen
POKE IT&+1,1  'Back Pen
POKE IT&+2,0  'Draw mode
POKEW IT&+4,0 'left
POKEW IT&+6,0 'top
POKEL IT&+8,0 'taPtr&
POKEL IT&+12,TextAddr&
POKEL IT&+16,0
ITTA&=IT&
END SUB

SUB MENUNUM (n%) STATIC
SHARED MenuPick
MenuPick=n% AND &H1F
END SUB

SUB ITEMNUM (n%) STATIC
SHARED ItemPick
ItemPick=INT(n%/32) AND &H3F
END SUB

SUB SUBNUM (n%) STATIC
SHARED SubPick
SubPick=INT(n%/2048) AND &H1F
END SUB

EndIt:
ClearMenuStrip& WINDOW(7)
LIBRARY CLOSE
SCREEN CLOSE 1
WINDOW CLOSE 2
MENU RESET
END
RETURN
```

•AC•

WHAT'S THE DIFF?

A review of Lattice's Compiler Companion

by Gerald Hull

Like many registered owners of Lattice's Amiga C Compiler, I recently received a notice in the mail concerning their "new" COMPILER COMPANION product. It features "10 important time-saving file-management tools designed for Amiga programmers". Many of these tools are indeed quite useful. However, similar programs also exist in the public domain. So the question is, are Lattice's versions worth the price?

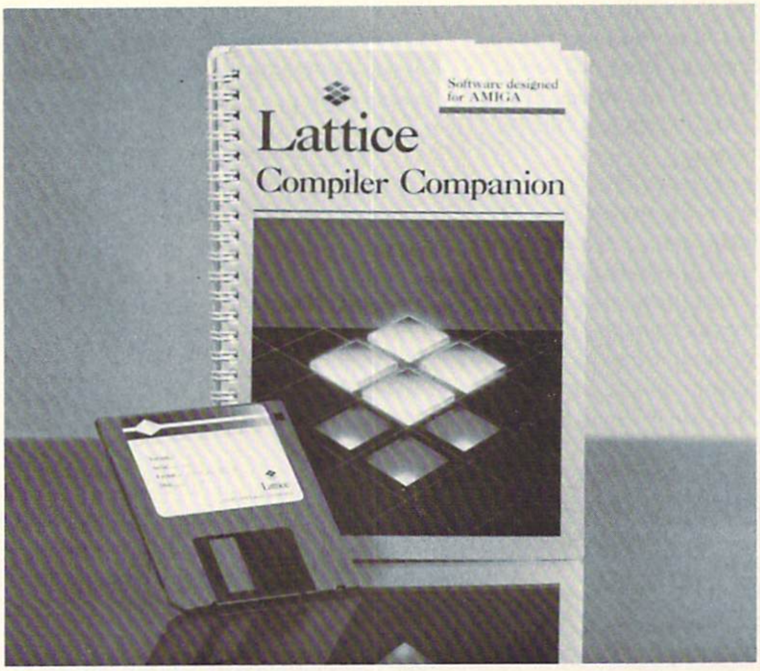
Program Management with LMK

I currently have a program under development whose source is distributed among seven C, four header, and four more assembler files. More than once I have wasted an entire day trying to find an apparent "bug" that actually was the result of accidentally linking in the wrong version of one of these files.

LMK is designed precisely to help programmers avoid this kind of costly mixup. The key to its function is the datestamp on the various starting (source), intermediate (object), and final (executable) files that together represent a program as a whole.

For each such program you (perhaps with a team of others) are working on, you prepare a command file which lists, for each file, the antecedents required to generate it. In addition, you specify the "actions" that need to be taken — compilation, assembly, linking, etc. — to generate each such "target" from those "dependencies."

So when you feed that command file, typically called "makefile" or "lmkfile" in the Lattice environment, to the LMK program, it checks to see if any of the direct or indirect dependencies of a program have a younger (or later) time and date. If so, it takes the indicated actions, thereby ensuring that those revised files are re-integrated into their intermediate and final targets.



For example, suppose you have a program, prog, whose source code is divided into three components — cpart1.c, cpart2.c and asmpart.asm — plus two header files, cpart.h and asmpart.i. Your makefile might look something like the following:

```
LCFLAGS = -b -r -iINCLUDE:

prog:  cpart1.o cpart2.o asmpart.o
     blink FROM LIB:c.o+cpart1.o+cpart2.o+asmpart.o \
TO prog LIBRARY LIB:lc.lib+LIB:amiga.lib

.c.o:
     lc $(LCFLAGs) $*

cpart1.o:  cpart1.c cpart.h

cpart2.o:  cpart2.c cpart.h

asmpart.o:  asmpart.asm asmpart.i
     assem asmpart.asm -i /include.i
```

The first line illustrates the "macro" capability of LMK: "LCFLAGS" acquires the value of the string following the equal sign. It is used later on in line 6, preceded by a dollar sign "$" and enclosed in parentheses to alert LMK that it is a macro. The next line has the executable file name, followed by a colon and its immediate antecedents in the development process.

The colon indicates that the target prog has cpart1.o, cpart2.o, and asmpart.o as dependencies. Consequently, if any of them have a newer datestamp than prog, LMK will perform the action specified. In this case, that action is an invocation of the blink linker, which gets passed along to the AmigaDOS operating system.

Note the use of the backslash "\" to indicate that the invocation is continued on line 4. The next line — ".c.o:" — illustrates a second use of the colon; this is called a "transformation rule." It says, that whenever LMK encounters a target which differs from its first listed dependency only in having a

(continued)

.o instead of a .c suffix, it should take the action indicated in line 6:

```
lc $(LCFLAGs) $*
```

This, of course, is a call to the C compiler, which indeed (in common usage) transforms .c files into .o files. We see here another use of the dollar sign, this time coupled with an asterisk: $*. This combination is a special symbol which LMK interprets to mean the target file name, stripped of any path specifications and of its suffix. Consequently, as LMK reads the dependency statements in lines 7 and 8, it knows it should perform the action specified by that transformation rule, if the file dates so require.

So, for example, if you (or one of your colleagues) have made changes in cpart1.c and asmpart.h since the last time prog was compiled, this makefile would result in asmpart.asm being reassembled and cpart1.c being recompiled. Finally, prog would be relinked from the new cpart1.o and asmpart.o object files, plus the old cpart2.o file (there being no need to recompile cpart2.c).

LMK has many more features, options, and capabilities. For example, if you invoke it with the "-t" option, instead of performing any actions, it will simply replace the datestamp of all files mentioned with the current time and date. You can do the same thing for individual files with the TOUCH program: By "touching" them, it brings them up to date.

Without going into all those features, you should by now have a good idea how the LMK can help to automate the program development process in an efficient and practical manner. Note that LMK will work just as well with Modula-2, Pascal or other high-level languages. Indeed, all of the COMPILER COMPANION programs, with the exception of CXREF, are general-purpose programming tools.

BUILD and EXTRACT

EXTRACT and BUILD are basically support tools for generating various kinds of command files quickly. Suppose you have a directory containing the files that are needed for a program. Let's simplify our earlier example, and suppose that the source for prog is contained in just two C language files, cpart1.c and cpart2.c.

In such a case, you may not feel that you need the full apparatus of LMK, but still you would like to generate an "executable" command file to direct the compilation process. First, you put the source components in their own directory, let's say, called "prog." EXTRACT now can be used to put the names of the files in that directory, each on a separate line, into a text file named "progtemp":

```
EXTRACT >progtemp -r prog/#?
```

The "-r" option indicates that path designators and suffixes should be stripped, and the use of the standard AmigaDOS wildcard, #?, indicates all the files in the prog directory should be included. Consequently, the contents of progtemp will be

```
cpart1
cpart2
```

The point of this becomes more obvious when we look at the functionality of BUILD. This latter tool typically takes input from two sources: A text file containing a list of file names (like that produced by EXTRACT), and the keyboard. Three special characters are distinguished in the keyboard input: The exclamation point, forward slash, and <ctrl> backslash.

Every line of keyboard input provided to BUILD represents a template that is repeated for each file name in the text file specified. In that template, "!" will be replaced by the file names, and "/" represents the newline character. (To get an ordinary forward slash, you need to repeat it twice: "//.") Finally, keyboard input is terminated with <ctrl> backslash. Suppose that, in our example, we would like to run each source component of prog through the stages of compilation separately. First, we would invoke BUILD:

```
BUILD >makeprog progtemp
```

The program is now awaiting keyboard input. So we give it

```
lc1 -b -iINCLUDE: -o!.q ! /lc2 -r -v !
 <ctrl>\
```

As a result, the newly created file makeprog will contain

```
lc1 -b -iINCLUDE: -ocpart1.q cpart1
lc2 -r -v cpart1
lc1 -b -iINCLUDE: -ocpart2.q cpart2
lc2 -r -v cpart2
```

Although it may seem a bit of a bother to learn to use BUILD and EXTRACT, they clearly will save time once mastered. There are many occasions when programmers need relatively repetitious command files, and these tools significantly simplify the process of creating them.

Regular Expression Parsing

Regular expressions are one of the "neat things" you learn about in computer science. They consist of those strings that can be defined in terms of literal characters plus the operations of concatenation and disjunction. The name of the GREP program, according to Lattice, stands for "Global Regular Expression Search and Print." I prefer to interpret it as "General Regular Expression Parser."

Regardless, the program represents a powerful tool for searching text files for specific kinds of character strings. If you are like me, you capture and save your sessions on networks like PeopleLink and BIX. Suppose, for some reason, you want to search a for all references to specific Amiga model numbers. If you use the AmigaDOS "search" function, you would have to issue at least three separate commands:

```
search capture/ "amiga 500"
search capture/ "amiga 1000"
search capture/ "amiga 2000"
```

However, even this wouldn't necessarily find all the references you are looking for, since sometimes there might be two spaces, or none instead of one, between the name and model number. GREP provides a much more powerful tool for such searches. You would only need to command

```
GREP "(Aa)miga *(512)00+" capture/#?
```

The search pattern is specified by the characters inside the double quotes (they wouldn't be needed if there were no spaces). Since the Lattice GREP, unlike "search," is case sensitive, we use brackets to indicate that the first letter can be both lower and upper case: [Aa]. This is how GREP accommodates disjunction: It looks for strings in which one, but only one, character matches what's inside the brackets.

Following the space after the computer name is another special symbol, the asterisk. This tells the program to look for zero or more instances of the preceding character (in this case, a space). Next we have sequence of numbers in brackets, since we are looking for either a 500, 1000, or 2000. If we wished, we could instead specify "[1-9]," where the dash allows us to represent the range of numbers between 1 and 9, thus broadening the search to include Amiga models as yet unborn and unnamed.

Our search pattern ends with two zeros followed by a plus sign. The latter is another special GREP symbol, telling it to look for one or more instances of the preceding character. There are many more such special characters. You can require, for instance, that the string occur at the beginning of a line ("^"), or at the end ("$"), and so forth. Here again there is a significant learning curve to overcome, but the reward is a powerful tool that will repay that effort many times over.

FILES and SPLAT

Indeed, these GREP-type search patterns are used by two other programs on the COMPILER COMPANION disk: FILES and SPLAT. The first is similar to AmigaDOS "list," but with additional functionality for copying and deleting files. It can use GREP patterns as well as AmigaDOS wildcards in determining which files to look for:

```
FILES -pat cpart(0-9)+\.c /cfiles
```

The backslash is used to indicate that the following character, a period, is to be taken literally. Otherwise, a period is interpreted as meaning any character whatever (printable or not).

The potential utility of the FILES command lies in its ability to search for file names recursively. That is, like "dir opt a", it will also look in subdirectories, subsubdirectories, and so forth. I qualify this with "potential", because, unfortunately, there are serious problems with FILES.

For one thing, the program contains a bug. With the COMPILER COMPANION disk in df1:, for example, the following command

```
df1:FILES -name w#? df1:
```

turns up nothing, even though the disk contains files named "wc" and "wc.c." When it does work, the program is very, very slow: it took 23 seconds to search a disk containing 32 files and only one subdirectory. Searching an entire hard disk will put it to sleep for a long time.

Another tool that uses GREP-type patterns is a little utility called SPLAT. It will perform search and replace operations on a text file or group of text files, without the need of first loading them into an editor (as you need to do with AmigaDOS "ed"). Lattice's documentation offers an example where you want to put a "-cc" on the end of every line that contains the expression "lc1" in the command file comp.bat:

```
SPLAT -dbak/ "lc1.*$" "& -cc" comp.bat
```

The "-d" option specifies a directory for the transformed version of comp.bat. SPLAT will not overwrite a file unless you explicitly tell it to. The ".*$" after "lc1" in the search string is an instruction to include everything that follows, up to the end of the line. Finally, the ampersand ("&") in the replacement string tells SPLAT to insert a copy of the search string. It's easy to imagine circumstances in which this program could be a helpful timesaver.

What's the DIFF

DIFF is a program that compares two text files and prints out their differences. It's very useful to a writer or programmer who wants to see the changes made in different versions of the same text. It has cropped up in a number of different computing environments: DEC RT-11, VAX VMS, CP/M, Unix, and now of course the Amiga.

LET YOUR WORK LOOK ITS BEST!

AMIGA Specialists

TRU-IMAGE™

Professional 35mm Slides

- Now you can have reproduction and presentation quality slides of your work
- Distortion-free--fills in raster lines crisp bright colors, converts all IFF files
- Photographic prints in various formats also available

Call (212) 777-7609 FOR DETAILS

Ask for Ilene--or write TRU-IMAGE
P.O. Box 660, Cooper Station
New York, N.Y. 10276

Amiga is a trademark of Commodore-Amiga

I first encountered it a long time ago in the Digital environment, and wasted a workday trying to figure out the algorithm. I later learned that there are many such algorithms, but when I got my Amiga, I decided it would be interesting and instructive to implement the one I had come up with.

Later I learned that Lattice had already released a DIFF, as well as various other folks in the public domain. My algorithm simply prints out the sections in which the compared texts differ. Lattice's prints out the changes you would need to make, one the same as the other, which means that the results will differ depending on the order in which you list the files being compared.

That is,

```
DIFF -p newfile oldfile
```

will generate the changes, insertions, and deletions necessary to make oldfile the same as newfile. And unlike my simple program, Lattice's DIFF recognizes of

(continued)

number of useful options. For example, the "-p" above screens out non-printable characters. You can filter out tabs and spaces (whitespace) as well, display the lines in common instead of those different, or restrict the range within the lines in which the comparison is performed.

WC and CXREF

Two final tools complete Lattice's COMPILER COMPANION. The first is a C code cross-reference generator name CXREF. As I mentioned earlier, this is the only tool in the package that is specific to the C language; all the others are equally suited to any high-level language, as well as other text processing tasks.

The basic output of CXREF, is a line-numbered listing plus a series of alphabetically ordered tables of preprocessor identifiers, functions, labels, structure identifiers, and (ordinary) identifiers, plus all the line numbers where they appear. You can run the program on a single file or on a whole directory full, with the appropriate use of AmigaDOS wildcards. For example,

```
CXREF >vd0:orbi.lst :l4/orbi/#?.c -iINCLUDE: -x
```

will produce a file named vd0:orbi.lst containing cross-reference tables for all the .c files in directory :l4/orbi.

Once again, a number of different options allow the user to finetune the results. For example, you can include the #include files in the processing ("-i" in the above example), generate listing or cross-reference table separately ("-x"), and control the format of the output. One thing that CXREF does not do, unfortunately, is indicate which occurrences of an identifier represent initial definitions, or assignments where its value can change. That sort of information can be very useful in tracking down bugs.

The final tool in the Lattice package is a modest little program entitled WC that will calculate the number of characters, lines and words in a text file. When I run it on the file containing this text, for instance:

```
WC latcc.txt
```

it prints out at the screen

```
Characters: 23134
Words: 3584
Lines: 475
```

Such information can be very useful. For instance if I am getting paid by the word, I can infer how much padding will be required to take care of my rent.

Critical Remarks

As you can see, Lattice's COMPILER COMPANION is a nice package of software development tools, which perform some very useful functions. But there are some flies in the ointment. The documentation is pretty bad, even by computer culture standards. Worse yet, in many places, it is inconsistent, vague, or incomplete.

The description of the "-b" option for EXTRACT first says "any filename extension (suffix) is ignored," but five pages later says "the file extension is to be included." The documentation of LMK seems like it was written for people who already know how it works, by someone who doesn't. A clear statement of LMK's datestamp dependent behavior does not appear until the appendix.

We never get a satisfactory discussion of the special characters SPLAT recognizes. Tabs, newlines, and ampersands are introduced in a backhand manner; in the case of the ampersand, one is referred back to a GREP discussion that doesn't exist. One wonders what else has been left out.

The user interface of the different components of the COMPILER COMPANION is less uniform than it could be. The options for the different tools are needlessly varied, making their fussy details harder to remember. Sometimes you need to redirect the screen output to get an output file (EXTRACT) and sometimes you need an "-o" option (DIFF). Sometimes you cannot redirect output even though you try (WC).

With GREP you need to specify "-V" to get the version number; all the other programs provide it automatically when you call them without arguments. And in general, the version numbers are odd. Programs that appeared on the TEXT MANAGEMENT UTILITIES disk as Version 1.01 Copyright 1984 show up on the COMPILER COMPANION disk as Version 1.00, Copyright 1988.

Does the bug in the CC version of FILES reflect the fact that it is more primitive than the bugfree 1.01 TMU version? Since the COMPILER COMPANION disk as a whole is labeled

Table One

LATTICE	PUB. DOMAIN	DISK	AUTHOR	SOURCE INCL'D
CXREF	XREF	Amicus #3	Philip T. Anstech	Yes
WC	WC	Fred Fish #69	Steve Summit	Yes
GREP	GREP	Amicus #1	DECUS	Yes
LMK	MAKE	Fred Fish #45	Hewlett-Packard	Yes
	MAKE	Fred Fish #69	Steve Walton	Yes
DIFF	DIFSSED	AM #11, FF #51	Chuck Forsburg	No
	DIFF	Fred Fish #87	Erik Baalbergen	Yes
		Fred Fish #107	D. Krantz	Yes
	DIFF	Fred Fish #142	Johan Widen	Yes
	DIFF	PeopleLink	Gerald Hull	No

HIGHER PERFORMANCE...AND CHEAPER TO BOOT!

FData-10 Single 3.5" External Drive **$149.95**
FData-20 Dual 3.5" External Drive w/Power Supply **$299.95**

- Fully 1010 Compatible
- Ultra Compact
- Acoustically Quiet
- Amiga® Color Coordinated
- Daisy Chainable
- Extra Long Cable
- High Performance
- Super Low Price

POLICY: Shipping and handling extra. Personal and company checks require 3 weeks to clear. For faster delivery, use your credit card or send cashier's check or bank money order. Credit cards are not charged until we ship. All prices are U.S.A. prices and are subject to change, and all items are subject to availability. These prices reflect a 5% cash discount. For all credit card purchases there will be an additional 5% charge. Defective software will be replaced with the same item only. All sales are final and returned shipments are subject to a restocking fee.

FLEXIBLE DATA
SYSTEMS, INC.
10503 FOREST LANE • SUITE 148 • DALLAS, TX 75243
FAX: 214-669-0021

214-669-3999

VISA MasterCard AMERICAN EXPRESS Card

Amiga® is a registered trademark of Commodore-Amiga, Inc.

Version 1.00, someone apparently decided all the programs should get the same number also. But can only confuse things to give presumably later versions of programs earlier numbers.

The Bottom Line

Still, those aren't hanging offenses, even though frustration with bad documentation may sometimes make you think otherwise. The tools on the COMPILER COMPANION disk can be very helpful if you take the time to master their intricacies. In particular, LMK and GREP are utilities I confess I had not previously taken the trouble to learn. They are well worth the effort.

The issue on the bottom line is whether the COMPILER COMPANION is worth the $100 list or approximately $75 discount price. In terms of previous Lattice practice, the evidence is ambiguous. On the other hand, the disk appears to be the combination of two previous Lattice disks, TMU and LMK, which sold for $75 and $125 respectively. This implies a savings of 50%, which isn't bad at all. On the other hand, I received the TEXT MANAGEMENT UTILITIES disk free with my upgrade to Lattice Amiga C Version 3.10.

This question is especially relevant because many of the programs on Lattice's CC have public domain equivalents. There are fine PD versions of LMK (usually called "MAKE"), DIFF, CXREF, WC, and GREP; see Table One. In general, Lattice's versions are somewhat more functional, for example, the indirect file capability of LMK. But this is not always the case.

The public domain DIF-SSED package, for example, can build a new file from the old, plus the list of differences generated, which helps to minimize phone bills. So if you feel you can do without BUILD, EXTRACT, a buggy FILES, and SPLAT, you probably can save yourself $70 to $100. (TOUCH exists as an option in the PD MAKE, as in LMK.)

On the other hand, Lattice is a reputable, well-established source of dependable Amiga software. They provide excellent support: A BBS, technical and customer service hotlines, and a conference on the BIX network. (You want to belong to BIX anyway for Amiga technical support, and Lattice helps out by including an $11.00 discount certificate.) If a public domain program messes up or acts funny, on the contrary, you usually have nobody to complain to but yourself.

Furthermore, Lattice gives you some other extras. You get "an unconditional 30-day, money-back guarantee" and of course are registered for update information. So far as software is concerned, the source code for WC is provided, plus useful code examples for using a library of GREP functions that is also provided. Then again, as you can see from Table One, almost all of the public domain equivalents come with complete source.

I would say that if you can afford the support, reliability, and extra functionality that Lattice provides, COMPILER COMPANION is worth the price. Otherwise, if your software shop is more economically oriented, you probably will be more than satisfied with programs in Amiga's rich and diverse pool of public domain software.

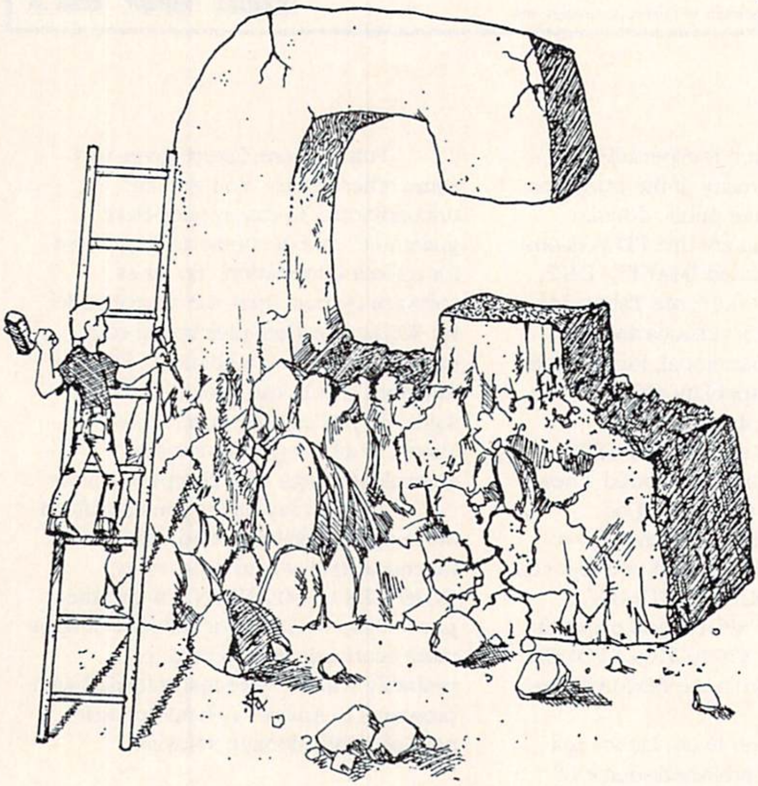
Disk Information

Version 1.00 of Lattice's COMPILER COMPANION comes on a single, unprotected disk, accompanied by a 154-page manual. It retails for $100, and is available from Lattice, Incorporated, 2500 S. Highland Avenue, Lombard, Illinois 60148; telephone 312-916-1600.

•AC•

On the Crafting of Programs

C Format



by David J. Hankins

Once again, welcome back. In this month's column, I'll take up the controversial topic of C format. Nothing, it seems, can stir up as much heated debate among C programmers than to suggest that they change the way their programs are formatted. So, when the fur begins to fly I wish to be prepared.

Don Full Battle Gear

Let's see - got my plate armor, got my shield, got my visored helm, got my gauntlets, even got my trusty longsword (I call him Stormbringer). Yeah, I think I'm ready to take on this month's topic. So without further digression, let's plunge ahead!

Aloof

One of things I like best about C is its syntax. I mean, what other language offers you the ability to write perfectly functioning, yet completely unintelligible (at least to those who don't yet know C) programs. In C you can let fly with cryptic symbols such as +, or ++, -, or —, =, or ==, |, or | |, & or && leaving those who don't know C scratching their heads in wonderment. And what about brackets? What other language offers you the choice of (, [, and {? Isn't this great! Knowing C we can write such nasty expressions as

```
for( x = 1, y = (int) foo( z.a[ 3 ] ) ; x < y ; x++ )
```

and leave our poor Basic friends in the dust. Kind of makes you feel smug, doesn't it?

Obfuscation

In all seriousness, however, the purpose of good format is to promote the ready comprehension of programs. Partially because of C's reliance on punctuation symbols, programs written in C can often be hard to understand. Good format can help remedy this.

As an aside, there is a yearly contest to see who can write the most unintelligible C program. Last year's winner of the 1987 International Obfuscated C Code Contest was Paul Heckbert. His entry is as follows:

```
#include <ctype.h>
#include <stdio.h>
#define _ define
#_ A putchar
#_ B return
#_ C index
char*r,c[30001],*d=">=<=!===||&&->++-
>><<",*i,*l,*j,*m,*k,*n,*h,*y;e,u=1,v,w,
f=1,p,s,x;main(a,b)char**b;{p=a>1?atoi(b[1]):79;r=c+read(0,j=l=i=c,30000);v=g(
j,&m);for(k=m;v!=2;j=k,m=n,v=w,k=m){w=g(k,&n);if(v==1&&m-
j==1&&*j==35)e&&A(10),
e=f=0;if(!f&&v==3&&(char*)C(j,10)<m)A(10),e=0,f=1;else
if(v>2&&(u||w)&&(f||u)&&
(l-i>1||*i!=61||n-k>1||!C("-*&",*k)))continue;else  if(v==3)if(f&&e+1+n-
k>p&&e)A
(10),e=0;else  A(32),e++;else{if(f&&e+m-j>p&&e)A(10),e=0;e+=m-
j;k=j;while(k<m)A(
*k++);}i=j;l=m;u=v;}e&&A(10);}g(j,m)char*j,**m;{if(j>=r)B*m=j,2;s=isdigit(*j)||
*j==46&&isdigit(j[1]);for(h=j;h<r;h++)if(!isalnum(*h)&&*h!=95&&(!s||*h!=46)&&(!
s||h[-1]!=101&&h[-1]!=69||!C("+-
",*h)))break;if(h>j)B*m=h,0;x=1;for(h=j;h<r&&C(
"  \t\n",*h);h++);if(h>j)h-
,x=3;if(*j==34||*j==39)for(h=j+1;h<r&&*h!=*j;h++)if(
*h==92)h++;for(y=d;*y&&strncmp(y,j,2);y+=2);if(*y)h=j+1;if(!strncmp("/
*",j,2)){
h=j+2;while(*++h!=42||*++h!=47);x=4;}*m=h+1;B  x;}
```

A Survey of Formats

Well, I don't think that any of us crank-out monstrosities like the one shown above, but, nevertheless, we may be writing our C programs in less than optimal format.

So what constitutes good format? And who can we look to for guidance on this matter? Since most people who know C possess the book "The C Programming Language" (frequently referred to as K & R) by Brian Kernighan and Dennis Ritchie perhaps this is as good a place as any to start. Using the format show in K & R, we would write the function tab() as follows:

```
/* Format shown in K & R */
tab(s,i)
char s();
int i;
{
        int l;

        l = strlen(s);
        if(i + l > 120)
          error();
        while (l < i){
          s(l++) = ' ';
        }
        s(l) = '\0';
}
```

This function appends space characters to the end of character string s until it reaches length i. Braces - you know what braces are don't you?; they're the little curly guys "{" and "}" - have purposely been used in the while construct to show the K & R brace format.

Another logical place to look for format guidance would be the C compiler manuals supplied by Lattice and Manx. Judging by the examples Lattice supplies, I would guess that the Lattice format for the tab() function would appear as follows:

```
/* Format shown in Lattice C Programer's Reference Manual */
tab(s,i)
char s();
int i;
{
        int l;

        l = strlen(s);
        if(i + l > 120) error();
        while (l < i)
          {
          s(l++) = ' ';
          }
        s(l) = '\0';
}
```

Note that Lattice moved the error() function call onto the same line as the if statement, and that braces for the while loop were placed on separate lines and were indented one tab set.

Now for Manx:

```
/* Format shown in Manx C source code (supplied with com-
mercial
package) */
tab(s, i)
char s();
int i;
{
        int l;

        l = strlen(s);
        if(i + l > 120)
          error() ;
        while (l < i) {
          s(l++) = ' ';
        }
        s(l) = '\0';
}
```

Notice Manx separates function arguments with an extra space (i.e. tab(s, i) versus tab(s,i)), the error() function call is on its own separate line, and braces used in the while construct do not appear on separate lines; instead, the opening brace is on the same line as the while statement and is preceded by a space, while the closing brace appears on its own line and is indented to the same level as the while statement.

Here's the format I've come to use. By the way, if you violently disagree with this format, remember - I'm wearing plate armor and have in my hands the sword of doom, Stormbringer (I knew this battle gear would come in handy!).

(continued)

```
/* Format based loosely on "Amended Proposal for C Code
Layout" */
tab( s, i )
char s( ) ;
int i ;
{
        int l ;

        l = strlen( s ) ;
        if( i + l > 120 )
          error() ;
        while ( l < i )
        {
          s( l++ ) = ` ` ;
        }
        s( l ) = `\0' ;
}
```

This format is a little more liberal with whitespace than some of the previous ones. Each semicolon is preceded by a space as are function arguments and binary operators. Like the Lattice format each brace is placed on a separate line, but unlike Lattice, braces are indented to the same level as the construct to which they belong. Like Manx, the error() function call was placed on its own line.

For those who are interested, the above format is loosely based on the "Amended Proposal for C Code Layout" which can be found in Dr. Dobb's Toolbook of C pp 44-47.

A Matter of Taste

Like many C programmers, I started out using the K & R format since this was the first text I had to draw upon for examples. And the K & R format is a good format - the code is reasonably easy to follow and whitespace is used to aid comprehension. But, if this format is so good, why did I change?

Well, the reason I eventually changed was that K & R seems a bit too skimpy with whitespace. I like to use lots of space - it seems to help me comprehend a program more readily (and as my friends will tell you, I need all the help I can get!). Also, I like to think of braces as marking off blocks of code. By placing braces on their own separate lines, the block nature of the code is emphasized. Besides, how many times have you left out a "}" in a program? It's not much fun staring at code for hours on end trying to pair each "{" with its corresponding "}". Believe me, placing each brace on a separate line makes finding the missing "}" a heck of a lot easier.

What it really boils down to, however, is personal taste. If you haven't decided on a C format you'd like to use, review each of the formats shown above. Which format was easiest for you to understand? Perhaps you like some of the "features" of one format and some in another. In this case, create your own format. However, whatever you decide to do, you should bear two things in mind:

1. Be consistent - don't mix formats within programs or you, and everyone else who has to look at your code, will no doubt end up helplessly lost.

2. Never try to force your chosen format upon others. C programmers are a proud lot and most are likely to defend their own format with something akin to religious fervor (by the way - if you insist on even bringing up this subject you may wish to invest in some medieval armament - works wonders at winning arguments).

So What's the Point?

So what's the point? Well, I guess the point of all this discussion is good format is critical to the understanding, and therefore to the debugging of code. I seem to recall reading somewhere that in the life of a program, 10% of the time is spent on initial program development, and 90% of the time on debugging and later modification. Given these figures, good format is not only critical, it is essential.

Next Month

In next month's column, I hope to discuss the new ANSI C "standard". Since Lattice already supports ANSI C, and Manx will support it with the 4.0 release of their compiler, it might be interesting to see what impact this might have on our C programming.

So until then,

```
main(){printf("Goodbye world!\n");}
```

er, that is

```
main()
{
  printf( "Goodbye world!\n" ) ;
}
```

Suggested Reading

If you'd like to explore this topic further, the following references may prove helpful:

Kernighan, Brian W. and Ritchie, Dennis M., *The C Programming Language.* Prentice-Hall, Inc., 1978.

The Editors of Dr. Dobb's Journal, *Dr. Dobb's Toolbook of C.* Brady, 1986.

About the Author

Having recently made the profound discovery that a block of data is 512 bytes long, and that compilers and linkers are beasts of an entirely different kind, Mr. Hankins is well on the way to complete enlightenment (yeah right). At times Mr. Hankins can be found on BIX [dhankins], Compuserve [76515,1650], or PeopleLink [OCS378].

•AC•

SYNC TIPS

DOT CRAWL

by Oran Sands

Welcome to Sync Tips, a monthly column that will serve up tasty morsels of byte-size information about video, graphics, the Amiga and how they work together. Every month, we'll cover a separate topic and try to explain some of the peculiarities of the video world. Please feel free to send in any questions, comments or answers you may have. If I don't know the answer, you can believe I'll find it!

Chroma Crawl

This month's topic is dot crawl (sometimes called "chroma crawl"). Dot crawl is one of the most annoying problems you'll see when using Amiga graphics with composite video devices. If you've ever looked at the vertical edge of a graphic, you'll see what I mean. The edge literally seems to be made of dots that are crawling upwards. It is most noticeable on the edges of saturated color graphics and is almost undetectible on graphics that are shades of gray. Hence, the term "chroma" crawl.

What is it and what causes it? Contrary to popular opinion, it's not due to any insufficiency on the part of the Amiga. It's a problem that's inherent in all NTSC composite video signals. Don't believe me? Watch HBO, Showtime or MTV and look at their graphics closely. You'll see every bit as much dot crawl. I recently used a broadcast color encoder worth $4000 to make a composite signal from my Amiga. The signal looked great but there was just as much dot crawl as the Amiga 1000's signal.

If we look at what causes dot crawl then we can better understand how to avoid it. In the standard NTSC video signal the chroma (color) portions of your signal are carefully intermixed with the luminance (brightness) information. Normally there would be no interference between the two but in real life, certain conditions can be created whereupon the luminance information can interfere with the chroma and cause the dot crawl effect. The actual physics of this are beyond the scope of this column but the conditions that cause it aren't. Whenever the luminance level of a graphic changes suddenly, say from a value of 0 to 9, then spurious signals are generated that cause the dot crawl to become evident. Don't forget that although the graphic may be blue or red it still has a brightness value associated with it. It is this sudden change from one level to another that is the cause of our problems.

Is it avoidable?

So how do we avoid dot crawl? Simply eliminate those abrupt changes in brightness. If you have a graphic with a vertical or slanted edge then find some way of gradually making the transition.

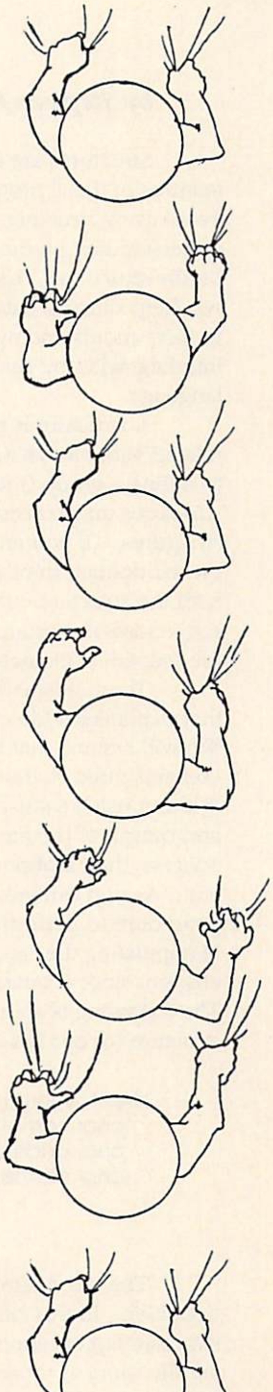
1. Using the BLEND or SMOOTH feature in most paint programs is an excellent way of doing this. You'll be anti-aliasing the edges as well as smoothing out the brightness transition.

2. Simply place a one pixel-thick line of a dimmer color (but the same shade) alongside the edge of the graphic. Then place another such line next to that one, this one a little dimmer yet. This also gives your graphics a very pleasing look.

3. Don't make your graphics all one color. Texture the graphics with other shades and colors. This can reduce, but may not eliminate, the dot crawl.

4. Since the problem is luminance transitions goofing up the color, then we can also control dot crawl by controlling our color. The problem of dot crawl is most pronounced when our graphics consist of very saturated colors like strong reds, blues, etc. In most paint programs you have control of your palette by two methods, R,G,B values or Hue, Saturation and Intensity values. By using the HSI controls you can study your colors and reduce the saturation levels. Normal television practice has never used a saturation level greater than 75% (a value of 12 to 13 on a scale of 0 to 15). This gives the dot crawl pattern less chroma to interfere with. Try to think pastels.

5. Reduce your colors to shades of gray. If reduced color is best, then no color is even better. Dot crawl simply doesn't show up in true shades of gray. Of course if this was all you needed you could have bought a Macintosh!



Although dot crawl is inherent in the NTSC video signal, there is really no reason you should have to watch it. A good television monitor or receiver with a comb filter can remove almost all of the dot crawl intereference. Unfortunately, these sets are often more expensive and won't be found in the average home. And remember, if you're designing your graphics for use on TV, then most of your viewers won't be watching on expensive TV's.

There are a number of such problems that are inherent in NTSC video signals. Dot crawl is one of the most annoying to users of computer graphics but there are others we need to watch out for. In the following months we'll cover a few of them. But first we'll talk about interlace, genlocks and other video/Amiga concerns. See you next month!

If you would like to contact the author please do so via this magazine or on PeopleLink's AmigaZone (ID- ojsands).

•AC•

C Notes

from the C Group

by Stephen Kemp ***Plink ID: SKEMP***

Structures are one of the many useful and powerful features of the C programming language. Some people try to avoid using structures, but I think this may be due to lack of understanding on the user's part rather than lack of useability on the structure's. Using structure variables in your programs will help simplify data manipulation and program maintenance. In fact, understanding structures is almost a must if you plan to interface with the Amiga's operating system using the C language.

A structure is simply a convenient way of referencing related variables as a single group. Unlike arrays that allow you to define a group (multi-dimension) of the "same" data type, structures may consist of multiple data types including other structures. (If you are familiar with the "record" data types that can be defined in other languages, then you are already familiar with the structure concept.) Although a structure can be referenced as a group, this does not prevent us from addressing the individual elements when necessary.

To understand structures, think about a simple program that maintains a file of names, addresses, and phone numbers. We will assume that each "record" (logical group) in the file contains three character arrays to hold each of these items. Without using a structure, we would have to handle the input and output of the file at the variable level—first a name, then an address, then a phone number, then a name, then an address, etc... As you can tell, this could become a very complicated procedure to perform if we had a more complex file definition. Manipulating the "records" of the file with a structure is more efficient since it treats the related information as a single unit. The following is an example of how we might represent the structure for our imaginary file.

```
struct record {        /* Record structure */
   char name(20);      /* The name area */
   char address(30);  /* The address area */
   char phone(10);    /* The phone number area */
};
```

The word "struct" is required to indicate a structure definition. In this example the word "record" is considered a structure tag or name. The rules for naming the structure tag are the same as those for naming variables. (Using the name "record" for our example is simply convenient and allows us to refer to this as the "record structure".) The individual components of the structure are placed between the open and close brace characters {}. As with all variable declarations, each element's declaration must be terminated with a semicolon. Since the structure is also a declaration, it too must be terminated by a semicolon after the closing brace.

This type of structure definition (with the structure tag) does not declare any actual data space, rather, it defines a "record" structure. (Remember that "record" is simply the name of this structure.) The following example shows how to declare a variable that does reserve data space using this structure definition.

```
struct record myrecord;   /* my file record variable */
```

Again, the word struct is required to tell the compiler the variable "myrecord" is a structure of the type "record". The compiler will set aside enough room in the data area to hold this record structure and reference it by the name myrecord.

There is a way to declare a structure variable that does declare data space without defining an intervening structure type. Using the same program example we could have declared the record variable like this:

```
struct {
   char name(20);     /* The name area */
   char address(30); /* The address area */
   char phone(10);   /* The phone number area */
}myrecord;               /* record structure */
```

As in the first example the word "struct" is still required. Notice, however, that no structure tag was indicated. Instead, a variable name is declared after the structure definition (after the closing brace and before the semicolon). The inclusion of the structure name (not tag) tells the compiler that space should be reserved for this variable.

Now your next question should be, "Why would I want to make a structure tag definition rather than just defining the structure variable that I want to use?" The answer has to be based upon what your program does and how it is written. Most programs will require several structure variables to be of the same "type". Without the structure tag, the definition of each variable would have to include the structure definition. With the structure tag, however, the definition of the structure resides in one place and you merely have to declare a variable of that structure type.

This means we don't have to declare the same structure over and over. Another advantage exists when it is necessary to alter the structure for some reason. Changing the structure tag definition will (after recompiling) change all the variables which use that definition. Other advantages can be found when it is necessary to "pass" structures to other functions or manipulate functions in different "modules" (when it takes more than one file to hold your C program).

Referring back to the phone number program, once we have a structure variable defined, the steps required to perform the file I/O become simplified. Reading and writing structures (which contains the three elements) is a much more effective way to maintain the data file since a single unit is being manipulated.

Addressing the individual elements of a structure variable requires the element's name as well as the structure's name. This is necessary because it indicates to the compiler which element of which structure is being referenced. For instance, the following statements would set the elements in our structure variable to John Doe, Nowhere Road, 555-1211.

```
strcpy(myrecord.name, “John Doe”);      /* name */
strcpy(myrecord.address,“Nowhere Road”); /* address */
strcpy(myrecord.phone,“555-1211”);     /* phone */
```

Notice how these statements address the individual elements of the structure. The structure's and elements' names are connected via a period (or "dot" as some programmers say). This dot tells the compiler that an element of the structure myrecord is being addressed, not the entire structure. Notice too, that these variables were defined as character arrays in our original definition. Since we omitted the index from each array, the compiler will assume they are address references (which is appropriate for this example). If a single element of the "name" array needs to be referenced, do it like any normal array by adding the index after the complete variable name.

```
myrecord.name(0) = ‘J’;  /* set element 1 of the name */
```

Naturally, the purpose of defining structures is to simplify the manipulation of related data. After the individual elements have been initialized to appropriate values, the entire structure will probably be "moved" somewhere else for further manipulation or I/O. This raises the questions of how to "pass" a structure as a parameter to another function or how to "assign" the values from one structure to another of the same type.

The answers to these questions will depend upon your C compiler and how "compatible" you want you program to be with other C compilers. Some compilers now have the technology to assign and pass structures by "value" (all the values of the elements), while many do not. Assigning a structure by value means that you could write a statement like the following (assume st1 and st2 are structures of the same type):

```
st1 = st2; /*assign the values of structure st2 to st1 */
```

D - Five Associates
19 Crosby Drive
Bedford, MA
01730-0523
(617) 275-8892
Tired of the high cost of computer repairs?
→FLAT Labor charges
→FREE Estimates
→Warranty work
Also:
1764 to 512K: $61.95
128 64K vdc RAM: $40.00
NEW: C=1902 conversion to RGB-I: $40.00
$19.95*
C-64/128 & peripherals
29.95*
Amiga 1000
$34.95*
Commodore PC-10
* plus parts and sales tax
Authorized Commodore Service Center

For the most part, it will be better if you avoid assigning and passing structures by value. This will help to ensure that no problems arise should you change compilers (or give your code away). Structures are normally handled by "address" and not by value. To indicate the address of a structure include the & character in front of the structure's name. Study the following example:

```
movmem(&st2,&st1,sizeof(st1)); /* st1 = st2 */
```

Assuming that "movmem" is a function that will move the contents from the address of parameter 1 (st2) to the address of parameter 2 (st1) for the length indicated in parameter 3 (sizeof(st1)). (The "sizeof" directive will instruct the compiler to determine the number of bytes required by this structure.) This function accomplishes the assignment of the previous example. As noted, the compiler "knows" to use the address of the structures because the address sign (&) preceded the structure name. I know that this may not be as straightforward as the simple "value" assignment, but it is compatible with all compilers (assuming the same definition of the function movmem). If your compiler supports structure assignments and you want to take advantage of that feature, then by all means do so.

Finally, we have to talk about how to receive and manipulate a structure when it is passed by "address" to a function. This is accomplished through the use of pointers. You may remember from my previous column about pointers to arrays, that having a pointer variable is almost like having the actual variable at your fingertips. Since that discussion, and in an effort to aide in your understanding of pointers to structures, I have come up with another "real life" example of a pointer.

Think about a television. That's right, a good ole' TV. Now when you want to change channels, do you get up from your sofa and walk over to punch the channel buttons, turn the volume up or down, or to turn it on or off? Of course not ... that is why remote controls were invented. With the remote you can do all these things from your chair without having to touch the TV set. In this sense then, a remote is a "pointer" to your TV. With a pointer you can manipulate the original without having to have the original defined locally.

(continued)

Look at this program that builds upon what we have already discussed and demonstrates the use of a pointer to a structure.

```
/* This program defines a record structure that contains */
/* a name, address, and phone number, calls a function to */
/* initialize the structure with some default values and  */
/* then prints the values to the console */

#include "stdio.h"                    /* include file for standard I/O */

struct record {        /* Record structure */
        char name(20);    /* The name area */
        char address(30); /* The address area */
        char phone(10);   /* The phone number area */
};

main()                               /* main function and program start*/
{
        struct record myrecord; /* record structure variable */

        set_record(&myrecord); /* set record to defaults */

        printf("Name - %s \nAddress - %s \nPhone - %s \n",
           myrecord.name,myrecord.address,myrecord.phone);
}

set_record(myrecord)     /* set the record to defaults*/
struct record *myrecord; /* a pointer to a record */
{
   strcpy(myrecord->name, "John Doe");       /* name */
   strcpy(myrecord->address,"Nowhere Road"); /* address */
   strcpy(myrecord->phone,"555-1211");       /* phone */
}
```

Focusing on the function "set_record", notice that when the parameter of a function is the address of a structure, we still have to use the "struct" declaration as well as the structure tag definition. Since this is a pointer to a structure, the asterisk (*) must precede the variable name.

Next, notice that unlike the example we used earlier in the column, the structure's and elements' names are not separated by a dot. Since this is not the actual structure (remember it is a pointer to a structure) we have to use a different symbol to indicate this to the compiler. The arrow (->) indicates that we are addressing an element of the structure "through" a pointer. (Actually, the arrow is a shorthand method for referencing elements via the pointer. If we wanted to manipulate the "name" variable without the arrow it would be written like this: (*myrecord).name. Most programmers (if not all) opt to use the "arrow" convention.)

This finishes our discussion of structures for now. Try typing in the example program, or writing your own program using structures. Keep working until you feel confident you understand them (don't forget to read about them in you reference materials). Remember a great deal of understanding comes with experience, so don't be afraid to try something new.

•AC•

Multi-Prefs

Time saving utilities for EVERY Amiga owner!

- Create multiple preference settings.
- Quickly and easily restore them as needed.
- Edit, Sort, Delete and Undelete any of them.
- Fast, Fun and Easy to use, from WorkBench or CLI.
- Runs on ALL Amigas . . . 500, 1000 and 2000 (V1.2 required).
- Multi-tasking.
- Not copy protected.
- Plus . . . 3 other time saving utilities are included.

Available NOW!!!

Just send $29.95 plus $2.00 S&H to:

SOFT·LINK inc. P.O. Box 304, Coventry, RI 02816

Dealer inquiries invited.
Amiga is a registered trademark of Commodore-Amiga, Inc.

STUFF-IT!

Design your own menus and use them with your favorite Amiga software!

- **Each menu is like having thousands of user definable, descriptively labeled, function keys**
- **Quickly and easily create all the menus you like**
- **Modify any of your menus, as often as needed**
- **Each menu can restore other Stuff-It menus**
- **Also included are a few menus we've created to get you started (CLI-Menus, ED-Menus, etc.)**
- **Runs on ALL Amigas...From WorkBench or CLI (V1.2 or higher required)**
- **No hardware requirements • NOT copy protected**

Stuff-It is a MUST for ALL Amiga owners, beginner to expert!

Just send $39.95 plus $2.00 shipping & handling to:

SOFT·LINK inc.

P.O. Box 304 Coventry, RI 02816

Amiga is a registered trademark of Commodore-Amiga, Inc.

Scrolling through SuperBitMap Windows

by Read Predmore

This article and its accompanying program, Scroll.c, describe how to implement SuperBitMaps for viewing or drawing into large graphic areas as well as moving them around with proportional scrolling gadgets. My curiosity about SuperBitMaps started with the Lines demo program on the workbench disk and my desire to utilize some of the same features in my graphics programming.

The following sections discuss creating SuperBitMaps, attaching them to a window, attaching horizontal and vertical scroll gadgets to the window and then interacting with the scrolling gadgets. A straight- forward drawing routine, patterned after the Lines demo program, is implemented as an example.

Overview of SuperBitMaps

A SuperBitMap is just a graphics work area that is usually, but not necessarily, larger than the window to which it is attached, and can be up to 1024 by 1024 pixels for the original Amiga graphics chips and at least 2048 by 2048 with the new ones. The SuperBitMap is created independently of any Screen or Window and its image can be retained while a screen is resized. An example use for a SuperBitMap would be to zoom in on a particular view of a MandelBrot calculation while being able to see the adjacent areas by scrolling them into view.

In the Scroll.c program, Listing 1, the function create_bitmap() is used to allocate the CHIP (graphics) memory for the SuperBitMap. First, memory for a BitMap structure is allocated and the Plane pointer array is set to zero. Then, the function tests as each bitplane is allocated and if there isn't enough room for the entire SuperBitMap the memory for the previous bitplanes is released by calling close_bitmap(). Note that the BitMap structure itself can be in either Fast or Chip memory.

When the entire BitMap is successfully allocated, its SuperBitMap pointer is passed to the create_window() function where it is stored in NewWindow.BitMap. The create_window() and create_screen() functions are derived from the make_window() and make_screen() functions from Inside the Amiga by J. T. Berry [Ref. 1], but create_window() has been embellished to include SuperBitMaps and gadgets. Note that the Window flags must call for a GimmeZeroZero window when a SuperBitMap window is used [Peck, Ref. 2].

Creating Proportional Scroll Gadgets

A good source of information on proportional gadgets is the set of articles by Harriet Tolly in Amazing Computing, Volume 2, issues 3, 5 and in particular issue 7, pages 72 to 80 [Ref. 3]. The book, Programmer's Guide to the AmigaTM, by R. A. Peck [Ref. 2] is another good source of information.

Each proportional gadget requires three structures for proper operation, the Image, PropInfo and Gadget structures. For default proportional gadgets such as are used in the window borders, the Image and PropInfo structures do not have to be initialized, but the Gadget structures Horiz_gadget and Vert_gadget are initialized. For the Horiz_gadget, the LeftEdge, TopEdge, Width and Height are 1, -7, -16 and 8 respectively, with similar numbers for the Vert_gadget. The negative numbers indicate that the TopEdge and Width are relative to the bottom and right-hand side respectively so that the gadget will resize properly when the window is resized.

The LeftEdge must be greater than zero for proper feedback of the gadget position. Under Intuition 1.2, if the LeftEdge is equal to zero the HorizPot member of the horiz_prop PropInfo structure does not reliably return a zero value when the gadget is pulled all the way to the left. The HorizPot value is used in the scroll_SBM() function to reposition the SuperBitMap with the ScrollLayer() function.

Pointers to the previously defined Image and PropInfo structures are stored in the GadgetRender and SpecialInfo member of the Gadget structure and the address of the Horiz_gadget is stored in the NextGadget member of the Vert_gadget to provide a linked list. Finally, the address of the Vert_gadget is passed to the create_window() function which stores this gadget address in NewWindow.FirstGadget.

Gadget and IDCMP Parameters

The gadget parameters include the Flags, Activation and GadgetType.

The flags are:

GADGHCOMP
Show gadget selection by complementing the colors.

GRELBOTTOM
Position the horizontal gadget relative to the bottom of the window.

GRELHEIGHT
Size the gadget relative to window height.

GRELRIGHT
Position the horizontal gadget relative to the right side of the window.

GRELWIDTH
Size the gadget relative to window width.

(continued)

Do you have a good idea for an article?

We can guide you
from **idea** to **published article**

Please Contact:

Editor
Amazing Computing
P.O. Box 869
Fall River, MA 02722
(508) 678-4200

The activation parameters are:

BOTTOMBORDER
Attach the gadget to the bottom border of the window.

FOLLOWMOUSE
Generate mouse position messages when this gadget is selected. Also requires the MOUSEMOVE flag for the window IDCMP port.

GADGIMMEDIATE
Immediately tell that this gadget has been selected by sending an IDCMP GADGETDOWN message. The IDCMP flag GADGETDOWN is also required.

RELVERIFY
When used with the IDCMP flag GADGETUP, a GADGETUP message is sent when the gadget selection is finished.

RIGHTBORDER
Attach the gadget to the right border of the window.

The GadgetType parameters are:

PROPGADGET
Designates a proportional gadget.

GZZGADGET
Signifies that the gadget is used with a GIMMEZEROZERO window.

Interacting with the Proportional Gadgets

The flags that are important for receiving messages about proportional gadget selection and repositioning are the FOLLOWMOUSE, GADGIMMEDIATE and RELVERIFY flags in the Activation member of the Gadget structure, and the IDCMP flags, GADGETDOWN, GADGETUP and MOUSEMOVE, which are set when opening the Window to which the Gadgets are attached.

When the Window is created or resized the variables dxmax and dymax in the main() function are recalculated. They define the maximum amount the screen can be scrolled and are determined by finding the width and height of the SuperBitMap and subtracting the width and height of the GimmeZeroZero window. The SuperBitMap size is constant but the GimmeZeroZero size will change when the window is resized. At the same time, the function update_scroll_gadgets() is used to recalculate the HorizBody and VertBody sizes in the horiz_prop and vert_prop PropInfo structures.

The proportional Gadgets have a GadgetID of HORIZ_SCROLL or VERT_SCROLL. These ID's are not system values but are defined in the scroll.c source code and are used by the function scroll_SBM(). The function scroll_SBM() positions the SuperBitMap within the window by using the HorizPot and VertPot settings to determine how much to move the SuperBitMap. Scroll_SBM() is called by three cases of input events; MOUSEMOVE, GADGETUP and NEWSIZE.

When the mouse is clicked on the gadget itself, the gadget_state variable is set to GADGETDOWN and the FOLLOWMOUSE activation in the Horiz_gadget and Vert_gadget causes IDCMP MOUSEMOVE messages to be sent. As long as the Gadget state is down the event loop will call scroll_SBM() to move the SuperBitMap along with the mouse movements.

The GADGETUP case occurs when the left mouse button has been clicked in one of the scroll gadget boxes. If the gadget itself was selected the MOUSEMOVE case has followed the mouse movements and no additional motion is necessary. However, if the mouse is clicked within the gadget box but not on the gadget itself, the gadget will be stepped towards the mouse and scroll_SBM() is called to reposition the SuperBitMap. The NEWSIZE case is used whenever the window is resized. The variables dxmax and dymax are recalculated as discussed above, update_scroll_gadgets() resizes the scroll gadgets, and scroll_SBM() repositions the SuperBitMap if necessary.

Prowling Around in the Amiga Data Structures

The Amiga data structures, while essential for the management of the Amiga's multitasking and windowed graphics display, present a labyrinth of structures nested within one another that is worthy of any Dungeon Master. I haven't conquered this maze, but have poked around in sub- dungeons 1 and 2 just enough to get things working smoothly.

I utilized several tools to find my way around. The first source of structure information is the Structure Browser program [Ref. 4], which is available on Fred Fish disk number 69. This program gives the real time values of Screen, Window and Gadget structures, etc., and was used to determine the SuperBitMap location within a Window structure and the gadget flags used for scrolling the SuperBitMap around in the Lines demo.

The second source of structure information is the Source level DeBugger (SDB) which is available as an accessory for the Manx Aztec C compiler version 3.6 [Ref. 5] and is one of the best debuggers available on any computer system. For example, a simple "p Window" command displays the entire Window structure with each member labeled. A "p *Window->RPort" shows the RastPort structure within the Window.

Drawing into the SuperBitMap

Once the SuperBitMap was attached to the window it was not clear where to point the Amiga graphics routines. Listing 2 is some output from SDB when it was used on the scroll.c program. Just after the SuperBitMap was created its parameters were printed out with the command "p *pSBM". It shows a BitMap with a size of 800 by 300 pixels. Next the Window structure is printed out and then the RastPort within the Window and the BitMap within that RastPort. (I told you this was a maze.) This BitMap is only 640 by 200 pixels and its Planes are not located at the same addresses as the Planes for the SuperBitMap. Finally, the original SuperBitMap was found within the WLayers member of the Window structure by printing "*pwindow0->WLayer->SuperBitMap".

Note that the actual output from the SDB program is not formatted exactly in this way. Comments have been added and the output has been moved around with appropriate indentation to emphasize the nesting of structures. Ellipsis (...) have been used to skip over structure members that are not relevant to this discussion.

There are two RastPorts in Listing 2, the RPort member of the Window structure and the rp member of the Window->WLayer, but it turns out that they both point to the same address. My conclusion was that the Amiga graphics routines draw into the SuperBitMap and then copy the visible portion into the Window->RPort for display in the window.

As an example of drawing into a SuperBitMap, the draw_lines() function creates patterns of lines which can roam over the full extent of the SuperBitMap. Although, only a portion of the SuperBitMap is visible at a time, any part of it can be viewed by using the horizontal and vertical scroll gadgets.

The draw_lines() function chooses the endpoints of lines to randomly fill the SuperBitMap using the random number generator, lran(), derived from Ref. [6]. A web of 32 lines is drawn between the current and previous line. Before the draw_lines() is finished the current line is stored in the previous line by swapping the end points. The pen color is incremented every fourth set of lines.

Makefile

The compiling and linking steps for Manx Aztec C version 3.6a are summarized in the makefile:

```
scroll: scroll.o
      ln +cb -g scroll.o -lc
      SDB Scroll

scroll.o: scroll.c
      cc -n scroll.c
```

The linker option +cb forces uninitialized data to be allocated in CHIP memory. The compiler option -n and the linker option -g are for the source level debugger.

An entire store dedicated to the commodore *AMIGA.*
Genlocks, Digitizers, Harddrives, Printers, Monitors, Modems, Memory Expansion, Books, and Magazines. Along with all of the latest software, in stock and on display.
As always customer satisfaction, and support come first!
Nothing but the best.

Remember...

The Memory Location

Where
Only *AMIGA* makes it possible.
We're Commodore specialists!
Only what works!

COMMODORE AMIGA

Open 10-6 Mon.-Thur. 10-8 Fri.
9-5 Sat. We ship UPS
Commodore authorized repair in store. Low flat rate plus parts.
Warranty service

The Memory Location
396 Washington St
Wellesley, MA 02181
(617) 237 - 6846

I hope that you find this useful. I can be contacted via Amazing Computing or on People Link as RPredmore.

References

John Thomas Berry, *Inside the Amiga,* Howard W. Sams & Co., Indianapolis, IN, 1986.

R. A. Peck, *Programmer's Guide to the Amiga,* Sybex, Alameda, California, 1987.

H. M. Tolly, *"Intuition Gadgets: .. the workhorses of Intuition"*, Amazing Computing, Vol. 2, No. 3, pp. 81-88, 1987.

H. M. Tolly, *"Intuition Gadgets: Boolean Gadgets"*, Amazing Computing, Vol. 2, No. 5, pp. 69-77, 1987.

H. M. Tolly, *"Intuition Gadgets: .. the workhorses of Intuition"*, Amazing Computing, Vol. 2, No. 7, pp. 72-80, 1987.

C. Zamara and N. Sullivan, *"The Transactor Amiga Structure Browser,"* Transactor, Vol. 7, Issue 6, May 1987, pp. 62-71.

Manx, Aztec C compiler and Source level DeBugger (SDB) Version 3.6a. Manx Software Systems, One Industrial Way, Eatontown, NJ 07724, Tel. (800) 221-0440.

W.H. Press, B.P. Flannery, S.A. Teukolsky and W.T. Vetterling, *Numerical Recipes in C,* Cambridge University Press, New York, 1987, Section 7.1, pp. 211.

(continued)

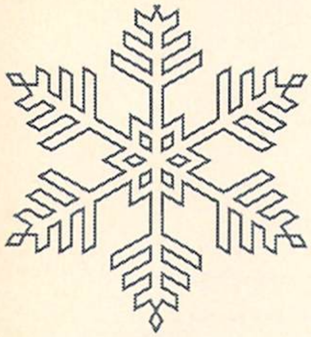
Makefile

```
########################################################################
#
#   Makefile to build SCROLL using Aztec C V3.6a for the AMIGA,
#
#   by Read Predmore,
#
#   21 July 1988.
#
########################################################################

scroll:  scroll.o
  ln +cb -g scroll.o -lc
  SDB Scroll

scroll.o:  scroll.c
  cc -n  scroll.c
```

Scroll.c

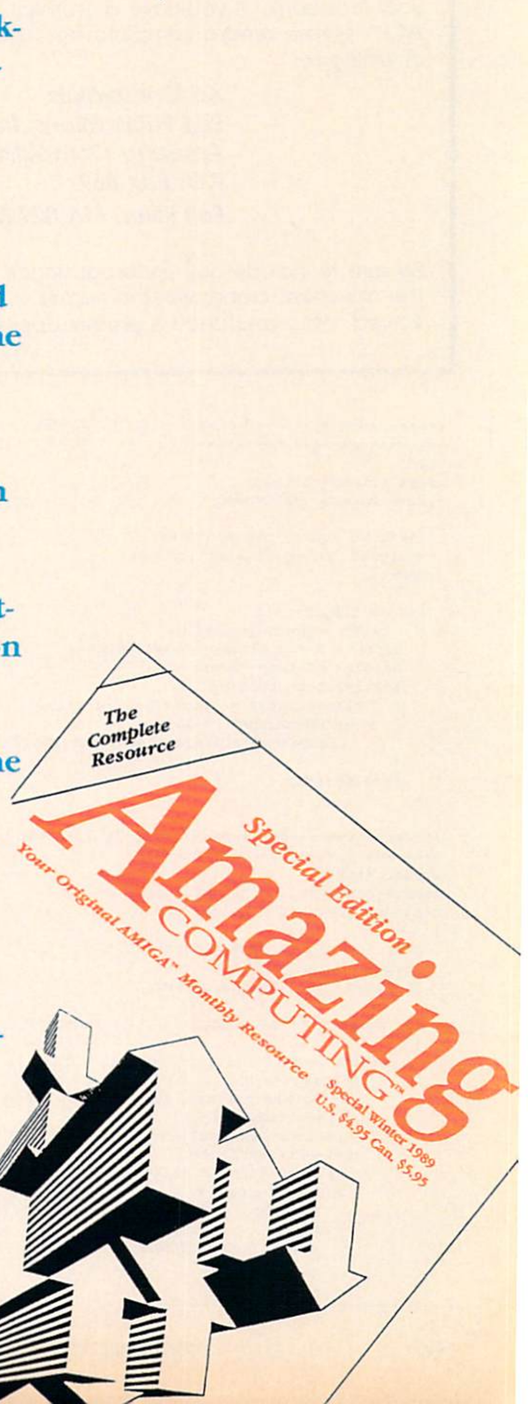
```
/*================================< SCROLL.C
>================================
 *
 *                      Copyright (C) 1988 by Read Predmore
 */

#include  <intuition/intuition.h>
#include  <exec/exec.h>
#include  <graphics/gfxbase.h>
#include  <stdio.h>
#include  <functions.h>

#define DOUBLE   double

/*  Resource Flags  */

#define F_INTUITION   0x000001L
#define F_GRAPHICS    0x000002L
#define F_LAYERSBASE  0x000004L
#define F_MATHTRANS   0x000008L
#define F_SCREEN      0x000020L
#define F_WINDOW0     0x000040L
#define F_SUPERBITMAP 0x001000L

#defineSCRN_HEIGHT  200
#defineSCRN_WIDTH     640
#define SCRN_DEPTH        3
        /*  SuperBitMap parameters.    */
#defineSBM_HEIGHT  300L
#defineSBM_WIDTH   800L
                                /*  IDs for scroll gadgets.    */
#defineHORIZ_SCROLL  0x0001
#defineVERT_SCROLL   0x0002

/*========================< FUNCTION DEFINITIONS
>========================*/
unsigned long   lran();
void    close_bitmap(), draw_lines(), init_color_map(), initialize(),
        *malloc(), myabort(), scroll_SBM(), update_scroll_gadgets();
struct  Screen  *create_screen();
struct  Window  *create_window();

/*========================< GRAPHICS DEFINITIONS
>========================*/
struct  GfxBase       *GfxBase;
struct  IntuitionBase *IntuitionBase;
struct  LayersBase    *LayersBase;

char    screen_title[80], window_title0[80];
long cur_resource;
struct  BitMap  *pSBM;
struct  Screen  *pscreen0;
struct  Window  *pwindow0;

struct Image   Horiz_image, Vert_image;
struct PropInfoHoriz_prop,  Vert_prop;

struct Gadget  Horiz_gadget =
{   NULL,
  1, -7, -16,  8,
  GADGHCOMP | GRELBOTTOM | GRELWIDTH,
  RELVERIFY | GADGIMMEDIATE | FOLLOWMOUSE | BOTTOMBORDER,
  PROPGADGET | GZZGADGET,
  (APTR) &Horiz_image,
  NULL, NULL, 0L,
  (APTR) &Horiz_prop,
  HORIZ_SCROLL, NULL
};

struct Gadget  Vert_gadget =
{   &Horiz_gadget,
  -15, 10,  16, -18,
  GADGHCOMP | GRELRIGHT | GRELHEIGHT,
  RELVERIFY | GADGIMMEDIATE | FOLLOWMOUSE | RIGHTBORDER,
  PROPGADGET | GZZGADGET,
  (APTR) &Vert_image,
  NULL, NULL, 0L,
  (APTR) &Vert_prop,
  VERT_SCROLL, NULL
};
Enable_Abort = FALSE;
/*  Global argument count which will be zero if started  from Workbench.
*/
int Gargc;
/*===============================< MAIN
>===================================*/
main(argc, argv)
int  argc;
char      *argv[];
{
  long dx, dy, dxmax, dymax;
  ULONGgadget_state;
  SHORTkeep_going = (SHORT) TRUE;
  USHORT  gadget_id;

  struct  IntuiMessage  myIntuiMessage, *tempIntuiMessage;
  struct  Window        *pcur_wind;

  cur_resource = 0;
  Gargc = argc;

  initialize();
  pcur_wind = pwindow0;
  dxmax = (8 * pcur_wind->WLayer->SuperBitMap->BytesPerRow)
             - pcur_wind->GZZWidth;
  dymax = pcur_wind->WLayer->SuperBitMap->Rows - pcur_wind->GZZHeight;
  gadget_state = GADGETUP;

  while(keep_going)
  {   draw_lines(pcur_wind);

     while(tempIntuiMessage = (struct IntuiMessage *)
          GetMsg(pcur_wind->UserPort))
     {   myIntuiMessage = *tempIntuiMessage;

        ReplyMsg(tempIntuiMessage);
        switch(myIntuiMessage.Class)
        {   case GADGETUP:
              gadget_id =
                 ((struct Gadget *) myIntuiMessage.IAddress)->GadgetID;
              scroll_SBM(pcur_wind, &Horiz_prop, &Vert_prop,
                      dxmax, dymax, gadget_id);
              gadget_state = GADGETUP;
              break;
           case GADGETDOWN:
              gadget_state = GADGETDOWN;
              gadget_id =
                 ((struct Gadget *) myIntuiMessage.IAddress)->GadgetID;
              break;
           case MOUSEMOVE:
              if(gadget_state == GADGETDOWN)
                 scroll_SBM(pcur_wind, &Horiz_prop, &Vert_prop,
                         dxmax, dymax, gadget_id];
              break;
           case NEWSIZE:
              dxmax = (8 * pcur_wind->WLayer->SuperBitMap->BytesPerRow)
                   - pcur_wind->GZZWidth;
              dymax = pcur_wind->WLayer->SuperBitMap->Rows -
                    pcur_wind->GZZHeight;
              update_scroll_gadgets(pcur_wind, &Horiz_prop, &Vert_prop);
              scroll_SBM(pcur_wind, &Horiz_prop, &Vert_prop,
                 dxmax, dymax, (USHORT) (HORIZ_SCROLL | VERT_SCROLL));
              break;
           case CLOSEWINDOW:
              while(tempIntuiMessage = (struct IntuiMessage *)
                   GetMsg(pcur_wind->UserPort))
                 ReplyMsg(tempIntuiMessage);
              keep_going = (SHORT) FALSE;
              break;
           default:
              break;
        } /* End of switch loop. */
     } /* End of while(IntuiMessage) loop. */
  } /* End of while(keep_going) loop. */

main_exit:;
  myabort("End of Scrolling through SuperBitMaps");
}
```

(continued)



AC's remedy for survival of the winter-time blues:

The winter season is upon us and with it brings not the anticipation of cold feet and joyous months of sub-zero tempuratures, but the arrival of a sizzling new item. Complete with an educational background, business knowledge, and an excitement for fun and games! Amazing Computing's Winter '89 Product Guide will keep you warmed up to the latest software and programming packages of the comming year. If you thought this winter would return along with bugs and the flu, *your right!* But we'll prescribe the Amiga remedies for all your ailments. Look for complete descriptions (and recommended dosages) for many software packages. From VIP--Virus Infection Protection which is a sure-fire cure for what makes your Amiga sick; to Hollywood Poker, guaranteed to heat up the atmosphere. You'll also enjoy a compilation of tutorials and referenced tittles authored by the experts who have helped shape the Amiga market. AC has taken the time to catalog over 1400 products for your Amiga, as well as a complete breakdown of the Public Domain Software Library. So keep this in mind when you can't seem to find a remedy for the winter-time blues. Turn to your Amiga with your best Amiga resource; Amazing Computing's Winter '89 Product Guide.



ATTENTION **READERS**

AC NEEDS YOU !

Amazing Computing™ cannot determine the dependability of advertisers from their advertisements alone. We need your feedback. If you have a problem with an advertiser in AC™, please send a complete description of the incident, in writing to:

Ad Complaints
PiM Publications, Inc.
Amazing Computing
P.O. Box 869
Fall River, MA 02722

Be sure to include any correspondence you have had with the advertiser, along with the names of the individuals involved. Your assistance is greatly appreciated.

```
/*===========================< CLOSE_BITMAP
>==========================*/
void
close_bitmap(pbitmap)
struct BitMap  *pbitmap;
{
  unsigned char  *planepointer;
  unsigned longdepth,width,height;
  shorti;

  if( pbitmap != 0L)
  {   depth = pbitmap->Depth;
     width = 8 * (pbitmap->BytesPerRow);
     height= pbitmap->Rows;
     for(i=0;i<depth;i++)
     {   planepointer =  pbitmap->Planes[i];
       if(planepointer != 0L)
          FreeRaster(planepointer,width,height);
     }
     free(pbitmap);
  }
}
/*===========================< CREATE_BITMAP
>=============================*/
struct BitMap *
create_bitmap(width,height,depth)
unsigned long width,height,depth;
{
  shorti, error_flag = (short) FALSE;
  struct  BitMap *pbitmap = 0L;
  unsigned char  *planepointer;

  if( (depth>0L) && (width>0L) && (height>0L) )
  {
     pbitmap = malloc(sizeof (struct BitMap));
     if( pbitmap != 0L)
     {   InitBitMap(pbitmap, depth,width,height);
        for(i=0;i<depth;i++)
           pbitmap->Planes[i]=(unsigned char *) 0L;
        for(i=0;i<depth;i++)
        {   planepointer = AllocRaster(width,height);
           if( planepointer )
              pbitmap->Planes[i] = planepointer;
           else
              error_flag = TRUE;
        }
```

```
     }
  } /* End of if (depth, etc.). */
  if(error_flag == TRUE)
  { close_bitmap(pbitmap);
     pbitmap = 0L;
  }
  return(pbitmap);
}
/*===========================< CREATE_SCREEN
>==========================*/
struct  Screen *
create_screen(topedge,width,height,depth, color0,color1, name)
short         topedge,width,height,depth;
unsigned char                              color0,color1;
         char                                                *name;
{
  struct  NewScreen ns;
  struct  Screen   *pscreen;
  ns.LeftEdge     = 0;
  ns.TopEdge      = topedge;
  ns.Width        = width;
  ns.Height       = height;
  ns.Depth        = depth;
  ns.DetailPen    = color0;
  ns.BlockPen     = color1;

/*  Mode options are HIRES, LACE, SPRITES and HAM which are defined
  in INCLUDE/GRAPHICS/view.h.  */
  ns.ViewModes    = NULL;
  if(width >= 640)
    ns.ViewModes |= HIRES;
  if(height >= 400)
    ns.ViewModes |= LACE;
  if( (depth == 6) && (!(ns.ViewModes & HIRES)) )
    ns.ViewModes |= HAM;

  ns.Type         = CUSTOMSCREEN;
  ns.Font         = NULL;
  ns.DefaultTitle = (UBYTE *) name;
  ns.Gadgets      = NULL;
  ns.CustomBitMap = NULL;
  pscreen = OpenScreen(&ns);
  return(pscreen);
}
/*===========================< CREATE_WINDOW
>==========================*/
struct  Window *
create_window(leftedge,topedge,width,height, color0,color1,
flags,IDCMPflags,
                pFirstGadget, window_title, pscreen, psbm)
short         leftedge,topedge,width,height;
unsigned char                              color0,color1;
unsigned long
flags,IDCMPflags;
struct Gadget  *pFirstGadget;
unsigned char                *window_title;
struct  Screen                             *pscreen;
struct  BitMap                                      *psbm;
{
  struct  NewWindow  nw;
  struct  Window    *pwindow;

  nw.LeftEdge    = leftedge;
  nw.TopEdge     = topedge;
  nw.Width       = width;
  nw.Height      = height;
  nw.DetailPen   = color0;
  nw.BlockPen    = color1;
  nw.Flags       = flags;
  nw.IDCMPFlags  = IDCMPflags;
  nw.Type        = CUSTOMSCREEN;
  nw.FirstGadget = pFirstGadget;
  nw.Title       = window_title;
  nw.CheckMark   = NULL;
  nw.Screen      = pscreen;
  if(flags & SUPER_BITMAP)
    nw.BitMap  = psbm;
  else
    nw.BitMap  = NULL;
  nw.MinWidth    = 200;
  nw.MinHeight   = 50;
  nw.MaxWidth    = pscreen->Width;
  nw.MaxHeight   = pscreen->Height;

  pwindow =  OpenWindow(&nw);
  return(pwindow);
}
```

```
/*===============================< DRAW_LINES
>==============================*/
#defineLRAN_MAX  259200L
void
draw_lines(pwind)
struct Window *pwind;
{
  static  int     idum = -1;
  static  long    color_count = 0,
                  color_times = 4,
                  pen_color = 0,
                  x1=0, x2=10,
             x3,x4,
             y1=0, y2=10,
             y3,y4,
                  dx12=10,  dx34,
             dy12=10,  dy34,
             npts = 32;
  long            i, x12,x34, y12, y34, xtemp, ytemp;
  unsigned longxmax, ymax;

  color_count = (color_count + 1) % color_times;
  if( ! color_count)
  {   pen_color = (pen_color + 1) % (1 << (pwind->RPort->BitMap->Depth));
    SetAPen( pwind->RPort, pen_color);
  }
  xmax  = 8 * pwind->WLayer->SuperBitMap->BytesPerRow;
  ymax  =     pwind->WLayer->SuperBitMap->Rows;
  x3 = (xmax * lran() ) / LRAN_MAX;
  y3 = (ymax * lran() ) / LRAN_MAX;
  x4 = (xmax * lran() ) / LRAN_MAX;
  y4 = (ymax * lran() ) / LRAN_MAX;
  dx34 = (x4-x3) / npts;
  dy34 = (y4-y3) / npts;

  for(i=0, x12=x1, y12=y1, x34=x3, y34=y3; i<npts; i++)
  {   Move(pwind->RPort, x12, y12);
    Draw(pwind->RPort, x34, y34);
    x12 += dx12;
    y12 += dy12;
    x34 += dx34;
    y34 += dy34;
  }
  x1 = x4;
  y1 = y4;
  x2 = x3;
  y2 = y3;
  dx12 = -dx34;
  dy12 = -dy34;
}
/*============================< INIT_COLOR_MAP
>================================
 *  Set up colors for the screen.
*/
void
init_color_map(pscreen)
struct  Screen  *pscreen;
{
  static short map_values[32] = {
   /*Format 0x0RGB            0x0RGB            0x0RGB            0x0RGB
*/
   /*  0 */ 0x0FFF , /*  1 */ 0x0000 , /*  2 */ 0x0F00 , /*  3 */ 0x000F ,
   /*  4 */ 0x00F0 , /*  5 */ 0x0FF0 , /*  6 */ 0x00FF , /*  7 */ 0x0F0F ,
   /*  8 */ 0x0444 , /*  9 */ 0x0888 , /* 10 */ 0x0CCC , /* 11 */ 0x0950 ,
   /* 12 */ 0x0090 , /* 13 */ 0x0FCA , /* 14 */ 0x02CD , /* 15 */ 0x0F0C ,
   /* 16 */ 0x0FFF , /* 17 */ 0x0000 , /* 18 */ 0x0F00 , /* 19 */ 0x000F ,
   /* 20 */ 0x00F0 , /* 21 */ 0x0FF0 , /* 22 */ 0x00FF , /* 23 */ 0x0F0F ,
   /* 24 */ 0x0444 , /* 25 */ 0x0888 , /* 26 */ 0x0CCC , /* 27 */ 0x0950 ,
   /* 28 */ 0x0090 , /* 29 */ 0x0FCA , /* 30 */ 0x02CD , /* 31 */ 0x0F0C
};
  LoadRGB4(&pscreen->ViewPort, map_values, 32L);
}
/*===============================< INITIALIZE >================================
 *  Open AMIGA libraries, screen and windows.                              */
void
initialize()
{
  unsigned char  color0, color1;
  unsigned long  flags, IDCMPflags;

  GfxBase = (struct GfxBase *)OpenLibrary("graphics.library",0L);
  if(GfxBase == NULL)
    myabort("Can't open graphics.library");
  cur_resource |= F_GRAPHICS;

   IntuitionBase=(struct IntuitionBase
*)OpenLibrary("intuition.library",0L);
  if (IntuitionBase == NULL)
    myabort("Can't open intuition.library");
  cur_resource |= F_INTUITION;

  LayersBase=(struct LayersBase *)OpenLibrary("layers.library",0L);
  if (LayersBase == NULL)
    myabort("Can't open layers.library");
  cur_resource |= F_LAYERSBASE;

  strcpy(screen_title,
        "Scrolling Through SuperBitMap Windows, Copyright 1988 by Read
Predmore");
  color0 = 0; color1 = 1;
  pscreen0 = create_screen(0, SCRN_WIDTH, SCRN_HEIGHT, SCRN_DEPTH,
                           color0, color1, screen_title);
  if(pscreen0 == NULL)
    myabort("Could not open the screen !");
    cur_resource |= F_SCREEN;
  ScreenToBack(pscreen0);
  init_color_map(pscreen0);

  pSBM = create_bitmap(SBM_WIDTH, SBM_HEIGHT, (long) SCRN_DEPTH);
  if(pSBM == NULL)
    myabort("Could not open the Super BitMap !");
    cur_resource |= F_SUPERBITMAP;

  Horiz_prop.Flags = AUTOKNOB | FREEHORIZ;
  Horiz_prop.HorizBody = 0x1FFF;
  Vert_prop.Flags = AUTOKNOB | FREEVERT;
  Vert_prop.VertBody = 0x1FFF;

  flags = WINDOWSIZING | WINDOWDRAG    | WINDOWDEPTH | WINDOWCLOSE |
       SUPER_BITMAP | GIMMEZEROZERO;

  IDCMPflags = MOUSEMOVE | GADGETDOWN | GADGETUP | CLOSEWINDOW | NEWSIZE;
  strcpy(window_title0,
        " SuperBitMap Window
");
  pwindow0 = create_window(0,10, SCRN_WIDTH,SCRN_HEIGHT-10, color0,color1,
                  flags,IDCMPflags, (struct Gadget *) &Vert_gadget,
                  window_title0, pscreen0, pSBM);
  if(pwindow0 == NULL)
    myabort("Could not open the window !");
  cur_resource |= F_WINDOW0;

  SetRast(pwindow0->RPort, 0L);
  update_scroll_gadgets(pwindow0, &Horiz_prop, &Vert_prop);
  ScreenToFront(pscreen0);
}
```

(continued)

Fisher's Computers and Software

714-987-1662 OR 987-8034

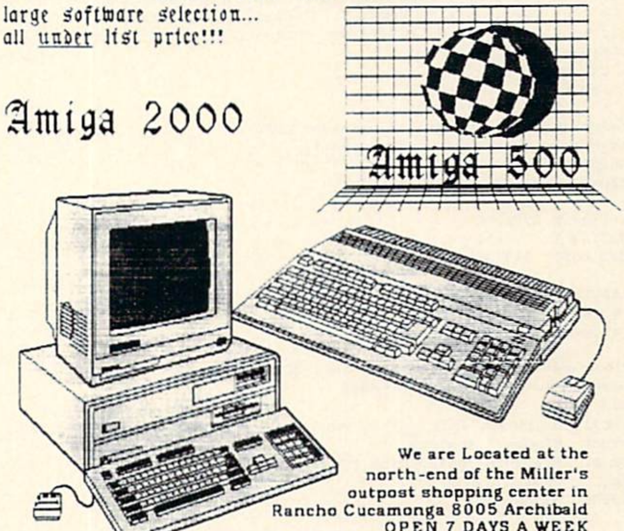
call for the latest prices

AUTHORIZED COMMODORE Amiga DEALER

both sales and service

large software selection...
all under list price!!!

Amiga 2000



We are Located at the
north-end of the Miller's
outpost shopping center in
Rancho Cucamonga 8005 Archibald
OPEN 7 DAYS A WEEK
Corner of Foothill Blvd. & Archibald Ave.

VISA

Amiga is a trademark of Commorore Amiga Inc.

```
/*==============================< MYABORT
>=================================
 *  Routine which closes windows and screeens and frees up memory which
 *  has been allocated for RasterPorts, etc.
*/
void
myabort(s)
char *s;
{
  if (cur_resource & F_WINDOW0)      CloseWindow(pwindow0);
  if (cur_resource & F_SUPERBITMAP) close_bitmap(pSBM);
  if (cur_resource & F_SCREEN)       CloseScreen(pscreen0);
  if (cur_resource & F_LAYERSBASE)  CloseLibrary(LayersBase);
  if (cur_resource & F_GRAPHICS)     CloseLibrary(GfxBase);
  if (cur_resource & F_INTUITION)   CloseLibrary(IntuitionBase);
  if(Gargc > 0)
    puts(s);
    exit(0);
}
/*================================< LRAN
>================================*/
unsigned long
lran()
{
#defineIM 259200L
#defineIA   7141L
#defineIC  54773L

  static unsigned long ulran  = 197;
  ulran = (ulran*IA + IC) % IM;
  return ulran;
}
/*===============================< SCROLL_SBM
>============================*/
void
scroll_SBM(pwind, horiz_prop, vert_prop, dxmax, dymax, gadget_id)
struct  Window   *pwind;
struct  PropInfo *horiz_prop,*vert_prop;
long    dxmax, dymax;
USHORT  gadget_id;
{
  long  dx = 0L,
       dy = 0L;
  if(gadget_id & HORIZ_SCROLL)
  {   dx = (dxmax * ((unsigned long) horiz_prop->HorizPot)) / 0xffffL;
     dx = dx - pwind->WLayer->Scroll_X;
  }
  if(gadget_id & VERT_SCROLL)
  {   dy = (dymax * ((unsigned long) vert_prop->VertPot)) / 0xffffL;
     dy = dy - pwind->WLayer->Scroll_Y;
  }
  if(dx || dy)
     ScrollLayer(pwind->WLayer->LayerInfo, pwind->WLayer, dx, dy);
}
/*==========================< UPDATE_SCROLL_GADGETS
>========================*/
void
update_scroll_gadgets(pwind, horiz_prop, vert_prop)
struct  Window       *pwind;
struct  PropInfo              *horiz_prop,*vert_prop;
{
  horiz_prop->HorizBody = (0xffff * ((unsigned long) pwind->GZZWidth)) /
                  (8 * pwind->WLayer->SuperBitMap->BytesPerRow);
  vert_prop->VertBody   = (0xffff * ((unsigned long) pwind->GZZHeight))
/
                 pwind->WLayer->SuperBitMap->Rows;
  RefreshGadgets(pwind->FirstGadget, pwind, 0L);

}
/*================================< FINI
>==================================*/
```

Structure Information from Source Level Debugger

```
print *pSBM               /*  SUPER_BITMAP  */
  struct BitMap =
  { unsigned int BytesPerRow = 100   /*  800 by 300 pixels. */
    unsigned int Rows = 300
    unsigned char Flags = 0
    unsigned char Depth = 3
    unsigned int pad = 64
     unsigned char *Planes[0] = 0x0002bf60
     unsigned char *Planes[1] = 0x00038760
     unsigned char *Planes[2] = 0x00044f60
  }
```

```
print *pwindow0
  struct Window =
  { struct Window *NextWindow = 0x00000000
    int LeftEdge = 0
    int TopEdge = 10
    int Width = 640
    int Height = 190
    int MouseY = 108
    int MouseX = 560
    int MinWidth = 200
    int MinHeight = 50
    unsigned int MaxWidth = 640
    unsigned int MaxHeight = 190
    unsigned long Flags = 1167
    struct Menu *MenuStrip = 0x00000000
    unsigned char *Title = 0x00013e64
    struct Requester *FirstRequest = 0x00000000
    struct Requester *DMRequest = 0x00000000
    int ReqCount = 0
    struct Screen *WScreen = 0x00234768
    struct RastPort *RPort = 0x00235ff8     /*  In FAST memory.  */
     /*  print *pwindow0->RPort  */
       struct RastPort =
       { struct Layer *Layer = 0x00234ff8
         struct BitMap *BitMap = 0x00234820
          /*  print *pwindow0->RPort->BitMap  */
             struct BitMap =
             { unsigned int BytesPerRow = 80 /* 640 by 200 */
               unsigned int Rows = 200
               unsigned char Flags = 0
               unsigned char Depth = 3
               unsigned int pad = 0
               unsigned char *Planes[0] = 0x000203e0
               unsigned char *Planes[1] = 0x00024260
               unsigned char *Planes[2] = 0x000280e0
             }
         unsigned int *AreaPtrn = 0x00000000
         struct TmpRas *TmpRas = 0x00000000
         ...
       }
    ...
    unsigned long IDCMPFlags = 627
    ...
    int GZZMouseX = 556
    int GZZMouseY = 97
    int GZZWidth = 618
    int GZZHeight = 171
    unsigned char *ExtData = 0x00000000
    char *UserData = 0x00000000
    struct Layer *WLayer = 0x00234ff8            /*  Window Layer  */
     /*  print *pwindow0->WLayer  */
       struct Layer  =
       { struct Layer *front = 0x00000000
         struct Layer *back = 0x00234e68
         struct ClipRect *ClipRect = 0x0020ec10
         struct RastPort *rp = 0x00235ff8
         struct Rectangle bounds =
         { int MinX = 4
           int MinY = 21
           int MaxX = 621
           int MaxY = 191
         }
         unsigned char reserved[0] = 0
         unsigned int priority = 999
         unsigned int Flags = 4
         struct BitMap *SuperBitMap = 0x00001998
          /*  print *pwindow0->WLayer->SuperBitMap  */
             struct BitMap  =
             { unsigned int BytesPerRow = 100
               unsigned int Rows = 300
               unsigned char Flags = 0
               unsigned char Depth = 3
               unsigned int pad = 64
               unsigned char *Planes[0] = 0x0002bf60
               unsigned char *Planes[1] = 0x00038760
               unsigned char *Planes[2] = 0x00044f60
             }
         struct ClipRect *SuperClipRect = 0x00235e88
         unsigned char **Window = 0x00232c38
         int Scroll_X = 0
         int Scroll_Y = 0
         ...
       }
    struct TextFont *IFont = 0x002006f8
  }
1
```

•AC•

by Forest W. Arnold

POINTERS, FUNCTION POINTERS, AND POINTER DECLARATIONS IN C

Introduction

In my article about linked lists, I promised to show how pointers to functions could be used to reduce data type dependencies. This article and the demo programs which accompany it, do just that. Listing 1 is a small program which shows how to declare and initialize function pointers, how to call a function using a pointer, and how to pass a function pointer to another function. Listing 2 is the output of the program. Listing 3 is a modified version of the linked list demo program which explains how to use function pointers with lists and shows how to declare and use complex pointer combinations.

The pointer is probably the most powerful feature of the C language, but is also the most difficult feature of the language to understand. Without pointers, C wouldn't be much of a language; with them, almost any programming task can be accomplished. After you learn to use pointers, you will be able to write programs which would be impossible to write in languages without pointers, and after you learn to use function pointers, you'll be able to write C programs which work like LISP programs, but faster.

Before jumping right into the subject of function pointers, I'll review what pointers are and how they are declared. First, I'll discuss pointers and simple variables, then pointers and arrays, followed by pointers and functions, and finally, complex pointer declarations. If you already know how to declare and use pointers, you will probably want to skip to the section on function pointers.

Pointers in General

A pointer is simply the address of some memory location in the computer or a 'virtual' memory location if the computer has virtual memory. In C, pointers are basic data types, and thus C contains two operators for manipulating the pointer type. These two operators are the '&' and the '*' operators. Given a variable name, the & operator will return the memory address of the variable. The & operator can be thought of as an 'address-of' operator. The * operator is called an 'indirection' operator and can be thought of as a 'pointer-to' operator. The '*' is used in two ways: it is used to declare a variable name as a pointer, and it is used to indirectly access the contents of a memory location.

Rules for using the & operator are that (1) the operand must be a variable, and (2) the operand cannot be an array name or a function name. The reason why the & cannot be used with array and function names is that both are already addresses.

The * operator can be used with any variable of any type. When used with a name in a declaration, the * operator declares that 'name' is a 'pointer-to' whatever 'type' appears in the declaration. For example, 'int *intptr' declares intptr to be a pointer to an int, and 'double *dblptr' declares dblptr as a pointer to a double. The important thing to remember about pointer variables is that the variable does not contain the data, but the address of the data; and until the variable is initialized, the address it contains is arbitrary (and thus garbage).

Since pointer variables contain only addresses, all pointer variables are the same number of bytes long (generally four bytes), and the address contained in the pointer variable is the address of the first byte of the pointed-to data. Thus pointers can be cast to any type. I generally use 'unsigned char *varName' as a 'typeless' pointer. However, before referencing the pointed-to data, it is important to cast the pointer to the correct type. This is because the compiler needs to know the type to determine how long the data is and how to generate offsets if the pointer points to an array or structure.

Pointers and Simple Variables

Suppose we have three variables declared as follows:

```
int intVal,newVal;
int *intPtr;
```

intVal and newVal are simple integer variables. *intPtr is a pointer to an int. Now suppose the compiler generated the following addresses for the variables:

name:	intVal	newVal	intPtr
address:	10000	11000	12000
contents:	?	?	?

The C code 'intVal = 10; newVal = 11; intPtr = &intVal;' will place values at the above addresses as follows:

name:	intVal	newVal	intPtr
address:	10000	11000	12000
contents:	10	11	10000

The variable intPtr now points to intVal's address, so intVal can be indirectly referenced through intPtr. The statement 'newVal = *intPtr;' causes the contents of the address

(continued)

pointed to by intPtr to be fetched and places the fetched contents in newVal's address. The table will now look like this:

name:	intVal	newVal	intPtr
address:	10000	11000	12000
contents:	10	10	10000

If you now use the statement '*intPtr = newVal * 3;', the value of newVal will be multiplied by three, the address stored in intPtr will be fetched, and the result of the multiplication will be stored at the fetched address. The table will then be:

name:	intVal	newVal	intPtr
address:	10000	11000	12000
contents:	30	10	10000

The chart below shows the syntax for accessing the addresses and contents of the variables in the table:

name:	intVal	newVal	intPtr
address:	&intVal	&newVa	&intPtr
contents:	intVal	newVal	intPtr
indirection:	—	—	*intPtr

As can be seen from the chart, the only difference between syntax for a pointer variable and a non-pointer variable is that the '*' operator can be used with a pointer to fetch the value stored at the pointed-to address.

Basically, only five things can be done with pointers: an address can be assigned to a pointer (intPtr = &intVal), the value at the pointed-to address can be accessed (newVal = *intPtr), the value at the pointed-to address can be replaced (*intPtr = newVal * 3), some arithmetic operations can be performed with pointers (below), and logical operations can be applied to pointers(ptr1 == ptr2).

Pointers and Arrays

As mentioned above, an array name is already an address, so it is illegal to use the & with an array name. With the declarations

```
int array(10);
int *arrayPtr;
```

' arrayPtr = &array;' is illegal, but either 'arrayPtr = array;' or 'arrayPtr = &array[0];' is perfectly legal and will assign the address of the first byte of 'array' to 'arrayPtr'. You may wonder why the statement 'arrayPtr = &array[0]' is legal, since I've already said the & operator can't be used with an array name. The answer is that 'array[0]' is not the name of an array, but the name of the first element in the array. It is because of little details like this that pointers are confusing.

After arrayPtr has been initialized to point to 'array', the following statements are equivalent:

	assign a value	access a value
array notation:	array(2) = 5	k = array(3)
pointer notation:	*(array+2) = 5	k = *(array+3)
pointer notation:	*(arrayPtr+2) = 5	k = *(arrayPtr+3)

Since the indirection operator has higher precedence than the addition operator, parenthesis must be placed around arrayPtr + n. The statement '*arrayPtr+2' will fetch the value stored at array[0] and add 2 to the value, but the statement 'arrayPtr+2' will generate a new address. When an integer n is added to an address, the value of n is automatically scaled by the correct number of bytes for the type of data being referenced. For example, if 'arrayPtr' points to address 10000 and the pointed-to type is a long integer, 'arrayPtr+1' will point to 10004. If the pointed-to type is double, 'arrayPtr+1' will point to 10008. This is why pointers have to be declared as pointing to a data type.

Since an integer can be added to or subtracted from a pointer, you will frequently see the '++' or '—' operators used with a pointer variable to iterate through an array. An example of this is the following code fragment:

```
int array(DIM);
int *arrayPtr;
for(arrayPtr = array; arrayPtr <= &array(DIM-1); arrayPtr++)
*arrayPtr += 10;
```

This fragment will loop through the array beginning with the first element and ending with the last element and add 10 to each array value. In the 'for' loop, arrayPtr = array sets arrayPtr equal to the address of the first array element, arrayPtr <= &array[DIM-1] compares the address stored in arrayPtr with the address of the last array element, and arrayPtr++ increments the address stored in arrayPtr. The statement *arrayPtr += 10 fetches the contents of the pointed-to address, adds 10 to the fetched value, then places the calculated value back into the pointed-to address.

Pointers and Functions

After the above discussion, function pointers are EASY. Function names are simply symbols which represent addresses tagged as a 'code' section. The compiler converts this symbol to an address, which is an offset from the beginning of the file in which the function is defined. The linker then uses the offset to create a new address for the function relative to the beginning of the load module. The loader then assigns an absolute address to the function when it is loaded. At least, I think that's how it all works. At any rate, declaring a variable as a pointer to a function and then getting the address of the function are the simplest pointer operations of all. The only thing to keep in mind is that all functions have to return some data type (or void), and the function pointer has to be declared as returning the same type as the function. If you have a function declared as 'double divideByTwo(number)', and you want to use a funtion pointer to execute the function, the way to declare the function pointer, assign it a value, then use it is:

```
double divideByTwo(),number,result;  /* regular stuff  */
double (*functionPointer)();         /* declare it     */
functionPointer = divideByTwo;       /* initialize it  */
result = (*functionPointer)(number); /* call it        */
```

The way to interpret 'double (*functionPointer)()' is 'a pointer to a function which returns a double'. Once again, because of the precedence of operators, '*functionPointer' must be enclosed inside parenthesis. Without the parenthesis, the declaration 'double *functionPointer()' would declare function-

(continued)

Pointer as 'a function returning a pointer to a double', which is something entirely different. And that's about all there is to function pointers!

The short demonstration program in listing 1 is an example of how to declare and use function pointers. The program has three functions: mulOp, divOp, and evalOp. mulOp simply multiplies two numbers and returns a pointer to the result. divOp divides two numbers and returns a pointer to the result. evalOp receives a function pointer and two numbers, and calls whatever function is pointed-to by the function pointer, then returns a pointer to the result.

A function pointer, 'binOp', is used as the pointer to either mulOp or divOp. It is set to point to either of these functions, which is then called from 'main' using the function pointer. Then, to demonstrate passing function pointers to other functions, mulOp and divOp are passed to evalOp as function pointers and evalOp calls the function using the pointer. Listing 2 contains a print-out of the function pointer data, the function addresses, and the results of the function calls.

Notice that each of these functions accept exactly the same number and types of arguments, and return the same type of data (an int defined as a BOOLEAN). Unfortunately, even when function pointers are used to avoid hard-coding function calls, the code still contains knowledge about the arguments which go into and are returned by the functions. This is not as severe a restriction as it first appears, because generally many functions within a program have the same calling protocol. This is especially true for interactive graphics programs. Functions which have the same argument protocol and calling sequence can be grouped into classes, and a function pointer can then be used to call any function within the class.

Even though the program in listing 1 is simple, the technique used could be applied in an interactive program which parses typed input, determines the desired operation, gets a pointer to the function for the operation from a table, then calls the function through the pointer. And speaking of tables of function pointers...

Complicated Pointer Declarations

Pointers can be combined with variables, arrays, structures, and functions to create all kinds of (almost) unreadable declarations. You can create tables of function pointers, pointers to functions which return pointers to other function pointers, function pointers inside structures, and many other combinations, some of which can be quite useful and most of which are quite confusing.

To compose and interpret complex declarations, you need to understand the precedence of the C operators and the order in which the operators are evaluated. I'll present only the precedence for operators which are commonly used in composing complex pointer definitions or which have been discussed. Each operator within the same group has equal precedence with the other operators in the group. The group of operators having the highest precedence are:

( ) function call
( ) array element
. structure or union member
-> structure or union member using a pointer.

This group of operators are evaluated from left to right. The group of operators having the next highest precedence are:

& address operator
***** indirection operator
++ increment operator
-- decrement operator
(others)

These operators are evaluated from right to left.

What all of this boils down to is that since the operators for functions and arrays have higher precedence than the indirection operator, parenthesis have to be used to declare a pointer to an array or function. Thus 'an array of pointers' is *ptrArray[DIM], but 'a pointer to an array' is (*arrayPtr)[DIM]; and 'a function returning a pointer' is *ptrFunc(), but 'a pointer to a function' is (*funcPtr)().

In 'C Programming Language - An Applied Perspective', authors Lawrence Miller and Alex Quilici present methods for composing and interpreting complex declarations. To compose the declaration, first write a description of it in English, with each part separated. Next, go through the description from left to right and find a C declarator for a variable of that type. Finally, go through the description again from left to right and replace each declarator for the identifier in the declarator to its right. I'll use declarations from the demo program in listing 3 to show how this method works.

The program stores function pointers in arrays dimensioned by the number of actions the program implements. Each action has a corresponding action procedure. For example, the search action is implemented by either searchLine() or searchPoint(). The program thus has an array which holds function pointers to the 'line' action procedures and an array which holds function pointers to the 'point' action procedures. These two arrays make up a table of function pointers indexed by the data type (line or point) and the action (search, delete, display, or displayInfo). Each action procedure returns an integer value of 0 or 1 indicating success or failure. Macros for BOOLEAN, FALSE, and TRUE are used for the function type and function return value.

Using the method described above, the function pointer array declaration is composed with the following steps:

1. ***English description -***
   an array of - pointers to - functions returning - BOOLEANs

2. *The first left to right scan produces:*
   funcPtr( )   *ptr   function( )   BOOLEAN

3. *Left to right substitution with parenthesis for precedence:*

   a. substitute funcPtr( ) for ptr -> *funcPtr()
   b. substitute *funcPtr( ) for function() -> (*funcPtr( ))( )
   c. append to base type -> BOOLEAN (*funcPtr( ))( )

(continued)

Amazing COMPUTING™

Reader Friendly - Reader Service

Wow, AC has done it again!
Due to the tremendous response of our original Reader Service Program, Amazing Computing announces an easier way to contact AC advertisers: The AC Reader Service Card Mailer.

To use the AC system, simply locate all the Amazing Advertisers you wish to contact below, find their appropriate AC Reader Service Card Numbers, and mark them on the card to the right. Fill in your address and mail the card. It is that easy!

The AC Difference
OK, this is not really a new idea. But, we have added a new twist. Gone are the awkward lines "Reader Service Card Number XXX" that reside somewhere on an advertisement. Instead, AC has listed all the advertisers in the Index Of Advertisers with their page numbers and special codes.

Now it is easy to find all the information you want for any AC advertiser in just one place. As always, Amazing Computing has placed their readers first!

Index of Advertisers

Advertiser	Page	Reader Service Number
Amazing Computer Systems	28	128
Antic Publishing	17	117
	37	137
Central Coast Software	43	143
Communications Specialties Inc.	46	146
D-Five Associates	99	199
Delphi Noetic Systems	83	183
Elipsys	79	179
Emerald Intelligence	49	149
Erich Stein & Associates	34	134
Expansion Technologies	51	171
EZ Soft	69	169
Fishers Computers and Software	107	210
Flexible Data Systems	93	193
Gramma Software	65	201
Great Valley Products	7	107
Great Valley Products (Dealers Only)	7	207
HI-Tech Graphics	35	135
Interactive Video Systems	53	153
King Publishing	70	170
	68	168
Lattice	13	113
Megatronics	CII	100
Micro Masters	52	152
Micro-Systems Software	55	155
Microbotics	19	119
Microsmiths	61	161
Micro Way	62	162
Mindware International	1	101
New Tek	CIV	200
New Wave Software	31	131
One Byte	56	156
OTG Software	54	154
Oxxi, Inc.	CIII	205
Poor Persons Software	30	130
Progessive Periphals	9	109
Quality Business Systems	65	265
Rainbow's Edge Production	87	187
Ronin Research and Development	4	104
Sedona Software	85	185
Soft Link	100	204
Software Advantage Consulting	60	160
Software Sensation	27	127
Software Terminal	14	114
Software Visions	29	129
The Bit Bucket Computer Store	76	176
The DISC COMPANY	21	121
The MEMORY LOCATION	103	203
The STERLING CONNECTION	57	157
The TRUMOR COMPANY	66	166
TRU-IMAGE	91	191

Rather than constantly having to figure out declarations in the code, C's 'typedef' can be used to simplify the program syntax and make it more understandable. In the demo program, the above array of function pointers is defined as type FUNCPTR_T. The actual declaration is:

```
typedef struct func
{
  BOOLEAN (*funcPtr(NUMACTIONS))( );
} FUNCPTR_T;
```

This type is then used to declare the table of function pointers as an array of two FUNCPTR_T structures:

```
FUNCPTR_T functionTable(2);
```

I'll give one more example of the technique for constructing complex declarations with another declaration from the demo program.

To manage the function pointer table, two procedures are used. One procedure simply initializes the table values. The other procedure accesses the table using the data type and the action code as indexes, and returns a pointer to the indexed function. A description of the access procedure is:

a function returning - a pointer to -
  a function returning -a BOOLEAN

The first left to right scan produces:

```
func( )        *ptr        func( )    BOOLEAN
```

left to right substitution then produces:

```
a.  *func( )
b              (*func( ))( )
c.                          BOOLEAN (*func( ))( )
```

The access function is named 'getFunction', so the final declaration is BOOLEAN (*getFunction())().

As you can see from the above, this technique for composing complex declarations is very useful and understandable. I was certainly happy to learn it. I knew that all types of declarations could be constructed in C, but didn't know how to make them until I read about the above method.

The method for converting a declaration to an English description is recursive: determine the type of an identifier by finding the type of the innermost declarator, write down the identifier type, substitute the identifier for its declarator, then continue the process until only a single identifier is left. C's operator precedence is used to determine the innermost declarator: it is an identifier followed by [] or (), or if neither of these is present, an identifier preceded by *. I'll use the two declarations from the demo program to show how this works.

The function pointer array, BOOLEAN (*funcPtr[])(), is decomposed as follows:

a. find innermost declarator - funcPtr( ) - an array
b. substitute 'x' for funcPtr( ) - BOOLEAN (*x)( )
c. find innermost declarator - *x - of pointers
d. substitute 'x' for *x - BOOLEAN x( )
e. find innermost declarator - x( ) - to functions returning
f. Only one identifier - BOOLEAN x( ) - BOOLEAN

The English description is thus 'an array - of pointers - to functions - returning BOOLEAN'.

The function declaration, BOOLEAN (*getFunction( ))( ), is decomposed in the same way:

a. innermost - getFunction( ) - a function returning
b. substitute 'x' for getFunction - BOOLEAN (*x)( )
c. innermost - *x - pointers to
d. substitute 'x' for *x - BOOLEAN x( )
e. innermost - x( ) - functions returning
e. Only one identifier - BOOLEAN x( ) - BOOLEAN

The English description of the declaration is then 'a function returning - pointers to - functions returning - BOOLEAN'. Magic!!! Now you should be able to easily construct and interpret just about any C declaration.

Applying Function Pointers

To show the usefulness of function pointers, the program in listing 3 is a version of the linked list demo program from my article about linked lists. The program demonstrates how function pointers can be used in place of hard-coded function calls to increase reusability of frequently used procedures, and shows one technique for managing the function pointers.

For those readers who missed the linked list demonstration program in my last article, the program graphically displays several lines and points in an intuition window and waits until an object is 'picked' with the mouse. Information about the picked object is then displayed in a text window, which remains displayed until the close gadget is selected. This continues until the graphical display window is closed, which ends the program. Pointers to the individual lines and points are stored in nodes in a linked list, and the entire program works by manipulating the list nodes. Each node contains a type code which is either POINT_TYPE or LINE_TYPE, and a generic (unsigned char *) pointer to the actual point or line structure.

There are several procedures for manipulating linked lists in the program. As originally written, some of the procedures were general-purpose, but several of them contained data type dependencies. The procedures searchList( ), displayList( ), displayNodeInfo( ), and deleteNode( ) had to determine the type of data pointed to by a node and then call either a 'point' procedure or a 'line' procedure to accomplish their purpose. As coded, the procedures could not be used for any type of data other than points and lines.

Linked lists are very useful creatures for interactive graphics programs where the user is constantly adding and deleting objects, so procedures which can deal with linked lists in a generalized manner without code modifications or code additions are also very useful. I'm a lazy programmer and don't like to rewrite the same basic procedures over and over, nor do I like adding to or modifying almost every procedure in a program when I add new data types or change existing data types. But how can this be avoided? You guessed it - with function pointers!

I mentioned above that most data types in graphics programs have basically the same procedures with the same argument lists, return values, and calling sequences. The linked list demo program from my last article is no exception: it has two data types, a point and a line, and each of these types has

(continued)

the same set of procedures with the same calling protocol. If arcs, circles, and rectangles were added to the program, they would also have the same set of procedures and calling protocol. This being the case, a single line of code with a function call via a function pointer can replace every specific function call in the list procedures and can also handle new data types. The only problem with using a function pointer is that some method of getting the correct pointer is needed.

There are two pieces of information available for each function call in the linked list procedures which can be used to select the correct function pointer. Each list node contains a type code for the pointed-to data, and each procedure implements a single specific action for a single data type. These two pieces of information can be used as indexes into a table of function pointers.

The type codes were defined in the original program. To use action implementions as an index, an action code is defined corresponding to each procedure. There are four action codes: SEARCH, DELETE, DISPLAY, and DISPLAYINFO. These correspond directly to the point and line procedure calls originally hard-coded in searchNode, deleteNode, displayNode, and displayNodeInfo.

I've already discussed the declarations used to construct the table of function pointers. With the action code indexes, here is what the table conceptually looks like:

action type	SEARCH	... ...	DISPLAYINFO
POINT_TYPE	searchPoint	... ...	displayPointInfo
LINE_TYPE	searchLine	... ...	displayLineInfo

Rather than store the table as a two-dimensional array, it is actually defined as an array of structures, with each structure corresponding to a row of function pointers for each type of data.

To simplify table creation and access and remove details about the table from the remaining code, the table is initialized in initFuncTbl() and accessed using (*getFunction())(). The initialization routine is called from main(). This procedure then calls bindPoint() and bindLine() with a pointer to each data type's function pointer array. The procedure bindPoint fills in the function pointers for the point data type and bindLine fills in the function pointers for the line data type. For example, inside bindPoint the code to initialize the function pointer for the point's search procedure is 'functionArray->funcPtr[SEARCH] = searchPoint'.

Although initFuncTbl contains type dependencies, the dependencies have all been removed from the four node procedures and placed in a single initialization procedure. Placing the type dependencies in a single procedure simplifies code modification and increases the reusability of the node procedures.

The access procedure (*getFunction())() is called with the data type from the list node and an action code. These two input arguments are used as function selectors by indexing into the function pointer table. The code which indexes the table is 'functionTable[type].funcPtr[action]' (remember that funcPtr is an array defined as a structure and functionTable is an array of funcPtr structures). An example of a call to the access routine (from deleteNode()) is 'delete = getFunction(node->dataType,DELETE)', where 'delete' has been declared as 'BOOLEAN (*delete)()'. Note that the procedure 'getFunction' is not really necessary.

The table can be directly accessed by any procedure in the program. Direct access into the table is certainly faster than calling a procedure to access it. The problem with doing this is that if the table data structure is changed, all of the procedures directly accessing it may also have to be modified.

The significant differences between this version of the list demo program and the previous version are that each data type now has a 'binding' procedure which links application specific code to general purpose code, two procedures for managing the linkages have been added, and type dependencies are no longer embedded in the linked list node procedures. The 'report' procedures in the original program which wrote information about the linked lists have been removed, and only one type of linked list (a LIFO list) is used in this version.

In a small demonstration program with only two types of data and which doesn't really do much, the impact of using function pointers probably isn't apparent. In a large application with 10 or more data types and 10 or more top-level actions, the reduction in the amount of code and the simplification of the code can be truly astounding.

Summary

Before I learned C, I had used a language which fully supported function pointers and recursive data definitions. I discovered how powerful and useful these language features were, and when I began learning C I wanted to know how to use the same features. Unfortunately, it was difficult to find examples of C programs which used function pointers, or to find textbooks which discussed complex declarations and function pointers in enough detail to be useful. So through trial and error (mostly error) and visits from the dreaded Amiga GURU, I figured out how to declare and use complex structures and function pointers. Hopefully, the material presented in this article and the demonstration programs have provided enough information for you to exploit C's more esoteric features while avoiding some of the pitfalls.

The following three statements summarize what has been discussed and demonstrated:

a. Pointers can be combined with C's basic built-in data types and with user-defined data types to construct almost any data structure imaginable.

b. Complex declarations in C can be methodically constructed and interpreted with a few simple rules.

c. Function pointers can be used to simplify code, reduce code volume, and increase the generality of procedures.

Further Reading

One of the best books I have found for learning C and which covers the material presented here in much greater detail is

C Programming Language - An Applied Perspective,
by Lawrence Miller and Alex Quilici,
published by John Wiley and Sons.

This book is available in paperback. It contains many clear examples of just about all of C's features and the best explanations of pointers I have seen anywhere.

Listing One
fpDemo1.c

```
/*=========================================================*/
/*  Listing 1 - function pointer demo, fpDemo1.c
 *
 *  simple demo of declaring, setting and using a
 *  function pointer.
 *
 *  Lattice compile & link command:  lc -Lm fpDemo1.c
 *
 *  (c)  Copyright 1988 by Forest W. Arnold
*/
/*=========================================================*/

#include <stdio.h>
#include <math.h>

#ifndef BOOLEAN
#define BOOLEAN   int
#endif
#ifndef FALSE
#define FALSE     0
#endif
#ifndef TRUE
#define TRUE      !FALSE
#endif

#define ABS(x) ( (x) < 0 ? -(x) : (x) )

/*
 *  declare some binary functions.
 *  These will all accept two double operands as input
 *  and a double pointer for output.  The function
 *  return value will be ‘TRUE’ for success, ‘FALSE’
 *  for an error condition (divide by zero)
*/

BOOLEAN mulOp();
BOOLEAN divOp();

/*  declare an ‘evaluate’ function                        */

BOOLEAN evalOp();

main()
{
   double   oper1,oper2;
   double   result;
   BOOLEAN  (*binOp)(); /* The function pointer         */

   binOp = NULL;  /*  set func ptr data to NULL         */

   printf(“\nFunction Pointer Demo...\n\n”);

   printf(“Function pointer data:\n\n”);
   printf(“  function pointer address:  %x\n”,&binOp);
   printf(“  function pointer contents: %x\n”,binOp);

   printf(“\nFunction addresses:\n\n”);
   printf(“  MUL function address:  %x\n”,mulOp);
   printf(“  DIV function address:  %x\n”,divOp);

   oper1 = 10.0;         /*  set test values            */
   oper2 = 15.0;

   /*
    *  Execute each function via its pointer
    *  First, set the function pointer to point to
    *     the address of the operator function.
    *  Next, output the function pointer.
    *  Next, call the function using the function ptr.
    *  Finally, output the results of the call
   */

   /*——  multiply using ptr to ‘mulOp’ ————*/

   binOp = mulOp;       /*  set function pointer      */
   printf(“\nMultiplying numbers. binOp = %x\n”,binOp);

   (void) (*binOp)(oper1,oper2,&result);  /* call it  */
   printf(“  %f * %f = %f\n”,oper1,oper2,result);

   /*——  divide using ptr to ‘divOp’ —————*/

   binOp = divOp;       /*  set function pointer      */
   printf(“\nDividing numbers. binOp = %x\n”,binOp);

   (void) (*binOp)(oper1,oper2,&result);  /* call it  */
   printf(“  %f / %f = %f\n”,oper1,oper2,result);

   /*
    *  Demo passing a function pointer to another
    *  function.  We’ll use same operands & perform
    *  same operations
   */

   /*——  pass multiply function & operands  —*/

   evalOp( mulOp,oper1,oper2,&result);
   printf(“Back from evalOp multiplication.\n”);
   printf(“   Product = %f\n”,result);

   /*——  pass divide function & operands  ——*/

   evalOp( divOp,oper1,oper2,&result);
   printf(“Back from evalOp division.\n”);
   printf(“   Dividend = %f\n”,result);

   exit(0);
}
/* mulOp - multiply operands & return ptr to result  */

BOOLEAN mulOp(oper1,oper2,result)
double   oper1,oper2;
double   *result;
{
   *result = oper1 * oper2;
   return( TRUE );
}
/* divOp - divide operands & return ptr to result    */

BOOLEAN divOp(oper1,oper2,result)
double   oper1,oper2;
double   *result;
{
   if ( ABS(oper2) <= 1.e-15 ) return( FALSE );

   *result = oper1 / oper2;
   return( TRUE );
}
/* evalOp - perform a binary operation via a func ptr */

BOOLEAN evalOp( operator,operand1,operand2,result )
BOOLEAN  (*operator)();       /* function pointer     */
double   operand1,operand2;   /* operands             */
double   *result;             /* function results     */
{
   if ( ! operator )    /* check func pointer         */
   {
      printf(“\nevalOp: NULL function pointer.”);
      return( FALSE );
   }
   else
   {
      printf(“\nevalOp:  function pointer = %x\n”,
               operator);
      return( (*operator)(operand1,operand2,result) );
   }
}
```

(continued)

Listing Two
Output of fpDemo1.c

```
=============================================
Listing 2.  Output of fpDemo1, simple
            function pointer demo
=============================================

Function Pointer Demo...

Function pointer data:

  function pointer address:  22a358
  function pointer contents: 0

Function addresses:

  MUL function address:  228d02
  DIV function address:  228d32

Multiplying numbers. binOp = 228d02
  10.000000 * 15.000000 = 150.000000

Dividing numbers. binOp = 228d32
  10.000000 / 15.000000 = 0.666667

evalOp:  function pointer = 228d02
Back from evalOp multiplication.
   Product = 150.000000

evalOp:  function pointer = 228d32
Back from evalOp division.
   Dividend = 0.666667
```

Listing Three
fpDemo2.c

```
/*=======================================================*/
/*  Listing 3.  Linked List Demo Program Source Code
 *
 *  fpDemo2.c - program to demonstrate generic linked
 *              lists with different data types using
 *              function pointers
 *
 *  lattice compile/link command:  'lc -L fpDemo2.c'
 *
 *  (c)  Copyright 1988 by Forest W. Arnold
*/
/*=======================================================*/

#include <exec/types.h>
#include <intuition/intuition.h>
#include <stdio.h>

#define INTUITION_REV 33
#define GRAPHICS_REV  33

struct IntuitionBase *IntuitionBase;
struct GfxBase       *GfxBase;

/*  'make life easy' stuff                                */

#define BOOLEAN   int
#ifndef FALSE
#define FALSE     0
#endif
#ifndef TRUE
#define TRUE      1
#endif
#ifndef NOT
#define NOT       !
#endif

#define NEW(x)    ( (x *)malloc( sizeof(x) ) )
#define FREE(x)   ( free( (char *)x ) )
#define ABS(x)    ( (x) < 0 ? -(x) : (x) )
#define MIN(x,y)  ( (x) < (y) ? (x) : (y) )
#define MAX(x,y)  ( (x) > (y) ? (x) : (y) )

/*
 *  define some data types
 *  We'll use a simple pixel point and a line
*/

typedef struct point
{
   short    x;
   short    y;
} POINT_T;

#define POINT_TYPE   0

typedef struct line
{
   short  x1,y1;
   short  x2,y2;
} LINE_T;

#define LINE_TYPE    1

/*  define a node for the linked list.                       */

typedef struct node
{
   struct node   *next;    /* the list 'link' pointer    */
   int           dataType; /* code for type of data      */
   unsigned char *data;    /* pointer to the actual data */
} NODE_T;

/*  define a list header for a LIFO list.                    */

typedef struct header
{
   NODE_T   *topNode;
} HEADER_T;

/*  define function selector action codes                    */
/*  These will be indexes into function array which will */
/*  'select' the function pointer                            */

#define SEARCH       0
#define DELETE       1
#define DISPLAY      2
#define DISPLAYINFO  3
#define NUMACTIONS   4

/*  define a function pointer array to hold pointers to  */
/*  functions returning 'BOOLEAN' values                  */
/*  using 'typedef' makes this easier to deal with        */

typedef struct func
{
   BOOLEAN  (*funcPtr[NUMACTIONS])();  /* func ptr array */
} FUNCPTR_T;

/* declare a function pointer table                          */
/* This declaration allocates two FUNCPTR_T arrays, one  */
/* for the POINT_T and one for the LINE_T                    */

FUNCPTR_T functionTable[2];

/* declare procedures for managing function pointers     */

BOOLEAN initFuncTbl();     /* initialize func ptr table  */
BOOLEAN (*getFunction())();/* return function pointer    */

/* declare the procedures for manipulating the data types*/

NODE_T  *createPoint();    /* return point & node        */
BOOLEAN bindPoint();       /* set 'point' function ptrs  */
BOOLEAN searchPoint();     /* look for point at x,y      */
BOOLEAN deletePoint();     /* delete point-node combo    */
BOOLEAN displayPoint();    /* graphic point display      */
BOOLEAN displayPointInfo();/* textual point display      */

NODE_T  *createLine();     /* return line & node         */
BOOLEAN bindLine();        /* set 'line' function ptrs   */
BOOLEAN searchLine();      /* look for line at x,y       */
BOOLEAN deleteLine();      /* delete line-node combo     */
BOOLEAN displayLine();     /* graphic line display       */
BOOLEAN displayLineInfo(); /* textual line display       */

BOOLEAN insertNode();      /* 'push' node onto node list */
BOOLEAN removeNode();      /* remove node from list      */
BOOLEAN deleteNode();      /* delete a node & data       */
BOOLEAN deleteList();      /* delete entire list         */
BOOLEAN displayList();     /* display entire list        */
BOOLEAN displayListInfo(); /* display info about a node  */
NODE_T  *searchList();     /* search list for a datatype */
```

```
/*  intuition interface routines                                */

BOOLEAN        openLibs();
void           closeLibs();
void           cleanUp();
struct Window *allocWindow();
void           gpDisplayMsg();
void           handleInput();

/*-------------------------------*/

main()
{
   struct Window  *window;
   HEADER_T        listHead;
   NODE_T         *node;
   unsigned long  wFlags;
   unsigned short iFlags;
   short          i,x,y,x1,y1;
   short          xInc,yInc,temp;

   wFlags = WINDOWCLOSE|WINDOWDRAG|SMART_REFRESH|ACTIVATE;
   iFlags = CLOSEWINDOW|MOUSEBUTTONS;

   /*
    *  initialize the list header.  Make sure list pointer
    *  starts out as ‘NULL’.
   */

   listHead.topNode = NULL;

   /*
    *  create the list, alternating between points and
    *  lines.
   */

   x = 120; y = 30;
   x1 = 135; y1 = 20;
   xInc = 40; yInc = 0;

   for ( i = 0; i < 5; i++ )
   {
      if ( node = createPoint( x,y ) )
         (void) insertNode( &listHead,node );

      if ( node = createLine( x1,y1,x1+xInc,y1+yInc ) )
         (void) insertNode( &listHead,node );

      x += 60; y += 20;
      x1 += 60; y1 += 20;
      temp = xInc;  xInc = yInc;  yInc = temp;
   }

   /* initialize the function pointer table                  */

   if ( NOT initFuncTbl() )
   {
      printf(“Can’t initialize function pointer table.\n”);
      cleanUp(&listHead,NULL);
      exit(1);
   }

   /* initialize intuition stuff & display the data          */

   if ( NOT openLibs() )
   {
      cleanUp(&listHead,NULL);
      exit(1);
   }
   if ( NOT (window=allocWindow(NULL,NULL,40,20,560,150,
                     “LIST DEMO”,wFlags,iFlags,-1,-1)) )
   {
      cleanUp(&listHead,NULL);
      exit(1);
   }
   SetAPen(window->RPort,1);
   Move(window->RPort,180,140);
   Text(window->RPort,“Pick entity to show data.”,25);
   (void)displayList( window,&listHead );

   /* process input until window close gadget picked         */

   handleInput(window,&listHead);
   cleanUp( &listHead,window );
   exit(0);
}
```

```
/*-------------------------------*/
void cleanUp(hdr,window)
HEADER_T      *hdr;
struct Window *window;
{
   if ( hdr ) deleteList( hdr );
   if ( window ) CloseWindow( window );
   closeLibs();
}
/*=======================================================*/
/*
 *  Procedures for managing function pointer table
 *  initFuncTbl() - call ‘bindPoint’ & ‘bindLine’ to set
 *                  function pointer array values
 *  getFunction() - return a pointer to a function
*/
/*=======================================================*/

BOOLEAN initFuncTbl()
{
   /*  call ‘binding’ routine for each datatype.  Send   */
   /*  each routine a pointer to its func ptr array      */

   if ( NOT bindPoint( &functionTable[POINT_TYPE] ) )
   {
      printf(“Can’t get pointers to ‘point’ routines.\n”);
      return( FALSE );
   }
   if ( NOT bindLine( &functionTable[LINE_TYPE] ) )
   {
      printf(“Can’t get pointers to ‘line’ routines.\n”);
      return( FALSE );
   }
   return( TRUE );
}
/*-------------------------------*/

BOOLEAN (*getFunction())( type,action )
int   type;    /* the datatype & the table ‘row’ index  */
int   action;  /* the action code & ptr array index     */
{
   if ( type < 0 || type > 1 ||
        action < 0 || action >= NUMACTIONS )
      return( NULL );

   return( functionTable[type].funcPtr[action] );
}
/*=======================================================*/
/*
 *  insertNode() - insert a node into a LIFO list
*/
/*=======================================================*/

BOOLEAN insertNode(hdr,node)
HEADER_T   *hdr;
NODE_T     *node;
{
   /*  check for valid hdr & node pointers              */

   if ( NOT hdr || NOT node ) return( FALSE );

   node->next = hdr->topNode; /* link it                */
   hdr->topNode = node;       /* make node the new ‘top’ */
   return( TRUE );
}
/*=======================================================*/
/*
 *  BOOLEAN deleteList() - delete an entire linked list
*/
/*=======================================================*/

BOOLEAN deleteList( hdr )
HEADER_T *hdr;
{
   NODE_T   *node;   /*  the current node to delete      */

   if ( NOT hdr ) return ( FALSE );

   while ( node = hdr->topNode ) /* stop when ‘NULL’     */
   {
      hdr->topNode = node->next; /* update list top ptr  */
      (void) deleteNode( node ); /* delete it            */
   }
   return( TRUE );
}
/*=======================================================*/
/*
 *  searchList() - look for a node at input coordinates
 *                 by calling ‘search’ function for the
 *                 datatype via a function pointer
*/
/*=======================================================*/
```

(continued)

```
NODE_T *searchList( hdr,x,y )
HEADER_T *hdr;
short    x,y;
{
   NODE_T   *node;         /* the current search node    */
   BOOLEAN  (*search)();   /* function pointer           */

   if ( NOT hdr ) return( FALSE );

   node = hdr->topNode;    /* start at top               */

   while( node )           /* stop when node is NULL     */
   {
      if ( search = getFunction(node->dataType,SEARCH) )
         if ( (*search)(node,x,y) ) /* call function     */
            return( node );         /* found the node    */

      node = node->next;   /* get next node in the list  */
   }
   return( NULL );         /* didn't find one at the x,y */
}
/*========================================================*/
/*
 *  displayList() - graphically display entire list
 *                  using func ptr to 'display' routine
*/
/*========================================================*/

BOOLEAN displayList( window,hdr )
struct Window  *window;
HEADER_T       *hdr;
{
   NODE_T   *node;         /* the current node           */
   BOOLEAN  (*display)();  /* ptr to display function    */

   if ( NOT hdr || NOT window ) return( FALSE );

   node = hdr->topNode;    /* start at top               */

   while( node )           /* stop when node is NULL     */
   {
      if ( display = getFunction(node->dataType,DISPLAY) )
         (void) (*display)( window,node );

      node = node->next;   /* get next node in the list  */
   }
   return( TRUE );
}
/*========================================================*/
/*
 *  displayNodeInfo() - text display of data for a node
 *                      using 'displayInfo' func ptr
*/
/*========================================================*/

BOOLEAN displayNodeInfo( window,node )
struct Window  *window;
NODE_T         *node;
{
   BOOLEAN  (*display)();  /* 'displayInfo' func ptr     */

   if ( NOT node || NOT window ) return( FALSE );

   if ( display = getFunction(node->dataType,DISPLAYINFO) )
      return( (*display)( window,node ) );
   else
      return( FALSE );
}
/*========================================================*/
/*
 *  removeNode() - remove a node from a FIFO list.
*/
/*========================================================*/

BOOLEAN removeNode(hdr,node)
HEADER_T *hdr;
NODE_T   *node;
{
   NODE_T   *prev,*curr;   /* previous,current nodes     */

   if ( NOT hdr || NOT node ) return( FALSE );

   /*  check the top node first                          */

   if ( NOT (prev = hdr->topNode) )
      return( FALSE );     /* list is empty              */

   if ( prev == node )     /* is it the top node ?       */
   {
      hdr->topNode = prev->next;    /* re-link nodes     */
      return( TRUE );
   }

   /* it wasn't the first one - have to search the list  */

   while( curr = prev->next )
   {
      if ( node == curr )              /* is this it ?      */
      {
         prev->next = node->next;      /* re-link nodes     */
         return( TRUE );
      }
      prev = curr;                     /* reset previous    */
   }
   return( FALSE );                    /* not in list       */
}
/*========================================================*/
/*
 *  deleteNode() - delete a single node & its data using
 *                 func ptr to 'delete' procedure
*/
/*========================================================*/

BOOLEAN deleteNode( node )
NODE_T   *node;
{
   BOOLEAN (*delete)();

   if ( NOT node ) return( FALSE );

   if ( delete = getFunction(node->dataType,DELETE) )
      return( (*delete)( node ) );
   else
      return( FALSE );
}
/*========================================================*/
/*
 *  Procedures for points:
 *  bindPoint() - initialize point's func ptr array
 *  createPoint() - create and initialize a point & its
 *                  list node
 *  deletePoint() - delete an instance of a point
 *  searchPoint() - see if a point is at input coordinates
 *  displayPoint() - graphically display a point
 *  displayPointInfo() - display data in message window
*/
/*========================================================*/

BOOLEAN bindPoint( functionArray )
FUNCPTR_T   *functionArray;
{
   if ( NOT functionArray ) return( FALSE );

   functionArray->funcPtr[SEARCH] = searchPoint;
   functionArray->funcPtr[DELETE] = deletePoint;
   functionArray->funcPtr[DISPLAY] = displayPoint;
   functionArray->funcPtr[DISPLAYINFO] = displayPointInfo;
   return( TRUE );
}
/*---------------------------*/

NODE_T *createPoint( x,y )
short    x,y;
{
   NODE_T   *node;   /* the list node to carry the point */
   POINT_T  *point;  /* the point structure              */

   if ( NOT (node=NEW(NODE_T)) )    /* allocate the node  */
      return( NULL );

   if ( NOT (point=NEW(POINT_T)) ) /* allocate the point */
   {                               /* didn't get it      */
      FREE( node );                /* release node memory*/
      return( NULL );
   }
   point->x = x;  point->y = y;    /* init the point     */

   /* initialize the node. Coerce point to unsigned char */

   node->next = NULL;
   node->dataType = POINT_TYPE;
   node->data = (unsigned char*)point;
   return( node );
}
/*---------------------------*/
BOOLEAN deletePoint( node )
NODE_T   *node;
{
   POINT_T  *point;

   /*
    *  make sure we have a valid node pointer, correct
    *  data type, and valid pointer to free
   */

   if ( NOT node || node->dataType != POINT_TYPE )
      return( FALSE );
   if ( NOT (point = (POINT_T*)node->data) )
      return( FALSE );

   FREE( point );
   FREE( node );
   return( TRUE );
}
```

```
/*------------------------------*/
BOOLEAN searchPoint( node,x,y )
NODE_T   *node;
short    x,y;
{
   POINT_T  *point;
   short    xDist,yDist;
   short    eps = 4;

   if ( NOT node || NOT (point = (POINT_T*)node->data) )
      return( FALSE );

   xDist = ABS( x - point->x );
   yDist = ABS( y - point->y );

   if ( xDist <= eps && yDist <= eps )
      return( TRUE );

   return( FALSE );
}
/*------------------------------*/
BOOLEAN displayPoint( window,node )
struct Window *window;
NODE_T         *node;
{
   POINT_T *point;
   short   x,y;

   /* make sure we have valid data, then display it      */

   if ( NOT window || NOT node ) return( FALSE );
   if ( NOT (point = (POINT_T *)node->data)) return(FALSE);

   x = point->x;  y = point->y;

   Move(window->RPort,x-2,y-2);
   Draw(window->RPort,x+2,y+2);
   Move(window->RPort,x-2,y+2);
   Draw(window->RPort,x+2,y-2);
   return( TRUE );
}
/*------------------------------*/
BOOLEAN displayPointInfo( window,node )
struct Window *window;
NODE_T         *node;
{
   POINT_T *point;
   char    str1[61],str2[61],str3[61];

   /* make sure we have valid data, then display it      */

   if ( NOT window || NOT node ) return( FALSE );
   if ( NOT (point=(POINT_T *)node->data)) return(FALSE);

   SetAPen(window->RPort,3);            /* highlight it    */
   (void) displayPoint( window,node );

   /* put data into strings & display message window     */

   sprintf(str1,"Node = %x, next = %x, data = %x",
               node,node->next,node->data);
   sprintf(str2,"Picked entity type is POINT_TYPE");
   sprintf(str3,"x = %d, y = %d",point->x,point->y);

   gpDisplayMsg(str1,str2,str3);     /* display the data  */

   SetAPen(window->RPort,1);         /* unhighlight it    */
   (void) displayPoint( window,node );
   return( TRUE );
}
/*=====================================================*/
/*
 *  Procedures for Lines:
 *  bindLine() - fill in line's function pointer array
 *  createLine() - create and initialize a line & its
 *                 list node
 *  deleteLine() - delete an instance of a line
 *  searchLine() - see if a line is at input coordinates
 *  displayLine() - graphically display a line segment
 *  displayLineInfo() - textually display line info
*/
/*=====================================================*/

BOOLEAN bindLine( functionArray )
FUNCPTR_T   *functionArray;
{
   if ( NOT functionArray ) return( FALSE );

   functionArray->funcPtr[SEARCH] = searchLine;
   functionArray->funcPtr[DELETE] = deleteLine;
   functionArray->funcPtr[DISPLAY] = displayLine;
   functionArray->funcPtr[DISPLAYINFO] = displayLineInfo;
   return( TRUE );
}
/*------------------------------*/
NODE_T *createLine( x1,y1,x2,y2 )
short    x1,y1;
short    x2,y2;
{
   NODE_T   *node;   /* the list node to carry the line  */
   LINE_T   *line;   /* the line structure               */

   if ( NOT (node=NEW(NODE_T)) )  /* allocate the node   */
      return( NULL );

   if ( NOT (line=NEW(LINE_T)) )  /* allocate the line   */
   {                              /* didn't get it       */
      FREE( node );               /* release node memory */
      return( NULL );
   }
   line->x1 = x1;  line->y1 = y1; /* init the line       */
   line->x2 = x2;  line->y2 = y2;

   /* initialize the node.  Coerce line to unsigned char */

   node->next = NULL;
   node->dataType = LINE_TYPE;
   node->data = (unsigned char*)line;
   return( node );
}
/*------------------------------*/
BOOLEAN deleteLine( node )
NODE_T   *node;
{
   LINE_T  *line;

   /*  make sure we have valid data                       */

   if ( NOT node || node->dataType != LINE_TYPE )
      return( FALSE );
   if ( NOT (line = (LINE_T*)node->data) )
      return( FALSE );

   FREE( line );
   FREE( node );
   return( TRUE );
}
/*------------------------------*/
BOOLEAN searchLine( node,x,y )
NODE_T   *node;
short    x,y;
{
   LINE_T   *line;
   short    minX,minY,maxX,maxY;
   short    eps = 4;

   if ( NOT node || NOT (line = (LINE_T*)node->data) )
      return( FALSE );

   /*
    *  NOTE:  this method of determining if point is on
    *  line only works for horiz & vert line segments
   */

   /*  build an 'epsilon' box around line segment         */

   minX = MIN(line->x1,line->x2) - eps;
   minY = MIN(line->y1,line->y2) - eps;
   maxX = MAX(line->x1,line->x2) + eps;
   maxY = MAX(line->y1,line->y2) + eps;

   if ( minX <= x && x <= maxX &&
        minY <= y && y <= maxY    )
      return( TRUE );

   return( FALSE );
}
/*------------------------------*/
BOOLEAN displayLine( window,node )
struct Window *window;
NODE_T         *node;
{
   LINE_T  *line;

   /* make sure we have valid data, then display it      */

   if ( NOT window || NOT node ) return( FALSE );
   if ( NOT (line = (LINE_T *)node->data)) return(FALSE);

   Move(window->RPort,line->x1,line->y1);
   Draw(window->RPort,line->x2,line->y2);
   return( TRUE );
}
/*------------------------------*/
BOOLEAN displayLineInfo( window,node )
struct Window *window;
NODE_T         *node;
{
   LINE_T  *line;
   char    str1[61],str2[61],str3[61];

   /* make sure we have valid data, then display it      */

   if ( NOT window || NOT node ) return( FALSE );
   if ( NOT (line = (LINE_T *)node->data)) return(FALSE);
```

(continued)

```
   SetAPen(window->RPort,3);           /* highlight line */
   (void) displayLine( window,node );

   /* put data into strings & display message window     */

   sprintf(str1,”Node = %x, next = %x, data = %x”,
               node,node->next,node->data);
   sprintf(str2,”Picked entity type is LINE_TYPE”);
   sprintf(str3,”x1 = %d, y1 = %d, x2 = %d, y2 = %d”,
           line->x1,line->y1,line->x2,line->y2);

   gpDisplayMsg(str1,str2,str3);    /* display the data  */

   SetAPen(window->RPort,1);        /* unhighlight line  */
   (void) displayLine( window,node );
   return( TRUE );
}
/*=====================================================*/
/*
 *  Intuition Utility Procedures:
 *  openLibs() - open libraries
 *  closeLibs() - close intuition libraries
 *  allocWindow() - allocate,inititalize & open window
 *  gpDisplayMsg() - display a message to a graphical
 *                   message window
 *  handleInput - intuition input handler
*/
/*=====================================================*/

BOOLEAN openLibs()
{
   IntuitionBase = (struct IntuitionBase *)
            OpenLibrary(”intuition.library”,INTUITION_REV);

   if ( NOT IntuitionBase ) return( FALSE );

   GfxBase = (struct GfxBase *)
              OpenLibrary(”graphics.library”,GRAPHICS_REV);

   if ( NOT GfxBase ) return( FALSE );
   return( TRUE );
}
/*——————————————————————————*/
void closeLibs()
{
   if ( IntuitionBase ) CloseLibrary(IntuitionBase);
   if ( GfxBase ) CloseLibrary(GfxBase);
}
/*——————————————————————————*/
struct Window *allocWindow( screen,gadget,x,y,w,h,name,
                            wFlags,iFlags,c0,c1 )
struct Screen  *screen;
struct Gadget  *gadget;
SHORT           x,y,w,h;
char           *name;
ULONG           wFlags;
USHORT          iFlags;
SHORT           c0,c1;
{
   struct NewWindow  NewWindow;
   struct Window     *window;

   NewWindow.LeftEdge = x;
   NewWindow.TopEdge = y;
   NewWindow.Width = w;
   NewWindow.Height = h;
   NewWindow.DetailPen = c0;
   NewWindow.BlockPen = c1;
   NewWindow.Title = name;
   NewWindow.Flags = wFlags;
   NewWindow.IDCMPFlags = iFlags;
   NewWindow.Type = WBENCHSCREEN;
   NewWindow.FirstGadget = gadget;
   NewWindow.CheckMark = NULL;
   NewWindow.Screen = screen;
   NewWindow.BitMap = NULL;
   NewWindow.MinWidth = 20;
   NewWindow.MinHeight =20;
   NewWindow.MaxWidth = 640;
   NewWindow.MaxHeight = 400;
   window = (struct Window *)OpenWindow( &NewWindow );
   return( window );
}
/*——————————————————————————*/
void gpDisplayMsg(str1,str2,str3)
char *str1,*str2,*str3;
{
   struct IntuiMessage *msg;
   struct Window       *msgWindow;
   unsigned long       wFlags,class;
   unsigned short      iFlags=CLOSEWINDOW|INACTIVEWINDOW;

   if ( *str1 == ‘\0’ && *str2 == ‘\0’ && *str3 == ‘\0’ )
      return;
   /*  allocate the message window                      */
   wFlags = WINDOWCLOSE|WINDOWDRAG|ACTIVATE;
   if ( NOT (msgWindow = allocWindow(NULL,NULL,50,20,550,
                  70,”= MESSAGE =”,wFlags,iFlags,-1,-1)) )
      return;

   if ( strlen(str1) > 60 ) /* clip txt len to 60 chars  */
      *(str1+60) = ‘\0’;
   if ( strlen(str2) > 60 )
      *(str2+60) = ‘\0’;
   if ( strlen(str3) > 60 )
      *(str3+60) = ‘\0’;

   SetAPen(msgWindow->RPort,1);  /*  display messages    */
   Move(msgWindow->RPort,10,20);
   Text(msgWindow->RPort,str1,strlen(str1));
   Move(msgWindow->RPort,10,30);
   Text(msgWindow->RPort,str2,strlen(str2));
   Move(msgWindow->RPort,10,40);
   Text(msgWindow->RPort,str3,strlen(str3));

   /* wait for message, then clear port & close window   */

   for(;;)
   {
      Wait( 1 << msgWindow->UserPort->mp_SigBit );

      while( msg = (struct IntuiMessage*)
                   GetMsg(msgWindow->UserPort) )
      {
         class = msg->Class;
         ReplyMsg( msg );

         if ( class == CLOSEWINDOW ||
              class == INACTIVEWINDOW )
         {
            while( msg = (struct IntuiMessage*)
                      GetMsg(msgWindow->UserPort) )
               ReplyMsg( msg );

            CloseWindow( msgWindow );
            return;
         }
      }
   }
}
/*—— ————————————————————————*/
void handleInput( window,hdr )
struct Window *window;
HEADER_T      *hdr;
{
   struct IntuiMessage *message;
   NODE_T              *node;
   unsigned short      mCode;
   unsigned long       mClass,iFlags;
   short               mX,mY;

   iFlags = window->IDCMPFlags; /* save IDCMP flags      */

   /* wait for a message to arrive                       */

   for (;;)
   {
      Wait( 1 << window->UserPort->mp_SigBit );

      /* process messages until there are none           */

      while( (message = (struct IntuiMessage *)
                        GetMsg(window->UserPort) ) )
      {
         /* get message info we'll need and process it   */

         mClass = message->Class;  mCode = message->Code;
         mX = message->MouseX;  mY = message->MouseY;
         ReplyMsg( message );

         /* see if this is an event we're interested in  */

         if ( mClass == CLOSEWINDOW )  /* shutdown & stop*/
         {
            while( (message = (struct IntuiMessage *)
                        GetMsg(window->UserPort) ) )
               ReplyMsg( message );

            ModifyIDCMP(window,NULL);      /* halt input  */
            return;
         }
         if ( mClass == MOUSEBUTTONS && mCode == SELECTUP )
         {
            /* search for entity & show entity info      */

            if ( node = searchList(hdr,mX,mY) )
            {
               ModifyIDCMP(window,NULL);  /* block input */
               (void) displayNodeInfo(window,node);
               ModifyIDCMP(window,iFlags);/* resume input*/
            }
         }
      }
   }
}
```

•AC•

The AMICUS & Fred Fish Public Domain Software Library

This software is collected from user groups and electronic bulletin boards around the nation. Each Amicus disk is nearly full, and is fully accessible from the Workbench. If source code is provided for any program, then the executable version is also present. This means that you don't need the C compiler to run these programs. An exception is granted for those programs only of use to people who own a C compiler.

The Fred Fish disk are collected by Mr. Fred Fish, a good and active friend of the Amiga.

Note: Each description line below may include something like 'S-O-E-D', which stands for 'source, object file, executable and documentation'. Any combination of these letters indicates what forms of the program are present. Basic programs are presented entirely in source code format.

AMICUS Disk 1

ABasic programs: Graphics

3DSolids	3d solids modeling prog. w/sample data files
Blocks	draws blocks
Cubes	draws cubes
Durer	draws pictures in the style of Durer
FScape	draws fractal landscapes
Hidden	3D drawing program, w/ hidden line removal
JPad	simple paint program
Optical	draw several optical illusions
PaintBox	simple paint program
Shuttle	draws the Shuttle in 3d wireframe
SpaceArt	graphics demo
Speaker	speech utility
Sphere	draws spheres
Spiral	draws color spirals
ThreeDee	3d function plots
Topography	artificial topography
Wheels	draws circle graphics
Xenos	draws fractal planet landscapes

ABasic programs: Tools

AddressBook	simple database program for addresses
CardFile	simple card file database program
Demo	multiwindow demo
KeyCodes	shows keycodes for a key you press
Menu	run many ABasic programs from a menu
MoreColors	way to get more colors on the screen at once, using aliasing
shapes	simple color shape designer SpeakIt speech and narrator demo

ABasic programs: Games

BrickOut	classic computer brick wall game
Othello	also known as 'go'
Saucer	simple shoot-em-up game
Spelling	simple talking spelling game
ToyBox	selectable graphics demo

ABasic programs: Sounds

Entertainer	plays that tune
HAL9000	pretends it's a real computer
Police	simple police siren sound
SugarPlum	plays "The Dance of the Sugarplum Fairies"

C programs:

ATerm	simple terminal program, S-E
cc	aid to compiling with Lattice C
decvnt	opposite of CONVERT for cross developers
Dotty	source code to the 'dotty window demo
echox	unix-style filename expansion, partial S,O-D
fasterfp	explains use of fast-floating point math
FixDate	fixes future dates on all files on a disk, S-E
freedraw	simple Workbench drawing prog.,S-E
GfxMem	graphic memory usage indicator, S-E
Grep	searches for a given string in a file with
docs.	ham shows off the hold-and-modify method of color generation
IBM2Amiga	fast parallel cable transfers between an IBM and an Amiga
Mandel	Mandelbrot set program, S-E
moire	patterned graphic demo, S-E
objfix	makes Lattice C object file symbols visible to Wack, S-E
quick	quick sort strings routine
raw	example sample window I/O
setlace	turns on interlace mode, S-E
sparks	qix-type graphic demo, S-E

Other executable programs:

SpeechToy	speech demonstration
WhichFont	displays all available fonts

Texts:

68020	describes 68020 speedup board from CSA
Aliases	explains uses of the ASSIGN command
Bugs	known bug list in Lattice C 3.02
CLICard	reference card for AmigaDOS CLI
CLICommands	guide to using the CLI
Commands	shorter guide to AmigaDOS CLI commands
EdCommands	guide to the ED editor
Filenames	AmigaDOS filename wildcard conventions
HalfBright	explains rare graphics chips that can do more colors
ModemPins	description of the serial port pinout
RAMdisks	tips on setting up your RAM: disk
ROMWack	tips on using ROMWack
Sounds	explanation of Instrument demo sound file format
Speed	refutation of Amiga's CPU and custom chip speed
WackCmds	tips on using Wack

AMICUS Disk 2

C programs:

alib	AmigaDOS object library manager , S-E
ar	text file archive program, S-E
fixobj	auto-chops executable files
shell	simple CLI shell, S-E
sq, usq	file compression programs, S-E
YachtC	a familiar game, S-E
Make	a simple 'make' programming utility, S-E
Emacs	an early version of the Amiga text editor, S-E-D

Assembler programs:

bsearch.asm	binary search code
qsort.asm	Unix compatible qsort() function, source and C test program
setjmp.asm	setjmp() code for Lattice 3.02
SVprintf	Unix system V compatible printf()
trees.o	Unix compatible tree() function, O-D

(This disk formerly had IFF specification files and examples. Since this spec is constantly updated, the IFF spec files have been moved to their own disk in the AMICUS collection.)

John Draper Amiga Tutorials:

Animate	describes animation algorithms
Gadgets	tutorial on gadgets
Menus	learn about Intuition menus

AMICUS Disk 3

C programs:

Xref	a C cross-reference gen., S-E
6bitcolor	extra-half-bright chip gfx demo, S-E
Chop	truncate (chop) files down to size, S-E
Cleanup	removes strange characters from text files
CR2LF	converts carriage returns to line feeds in Amiga files, S-E
Error	adds compile errors to a C file, S
Hello	window ex. from the RKM, S
Kermit	generic Kermit implementation, flakey, no terminal mode, S-E
Scales	sound demo plays scales, S-E
SkewB	Rubik cube demo in hi-res colors, S-E

AmigaBasicProgs(dir)

Automata	cellular automata simulation
CrazyEights	card game
Graph	function graphing programs
WitchingHour	a game

ABasiC programs:

Casino	games of poker, blackjack, dice, and craps
Gomoku	also known as 'othello'
Sabotage	sort of an adventure game

Executable programs:

Disassem	a 68000 disassembler, E-D
DpSlide	shows a given set of IFF pictures, E-D
Arrange	a text formatting program, E-D

Assembler programs:

Argoterm	terminal program with speech and Xmodem, S-E

AMICUS Disk 4 **Files from the original Amiga Technical BBS**

Note that some of these files are old, and refer to older versions of the operating system. These files came from the Sun system that served as Amiga technical support HQ for most of 1985. These files do not carry a warranty, and are for educational purposes only. Of course, that's not to say they don't work.

Complete and nearly up-to-date C source to 'image.ed', an early version of the Icon Editor. This is a little flaky, but compiles and runs.

An Intuition demo, in full C source, including files: demomenu.c, demomenu2.c, demoreq.c, getascii.c, idemo.c, idemo.guide, idemo.make, idemoall.h, nodos.c, and txwrite.c

addmem.c	add external memory to the system
bobtest.c	example of BOB use
consoleIO.c	console IO example
creaport.c	create and delete ports
creastdi.c	create standard I/O requests
creatask.c	creating task examples
diskio.c	example of track read and write
dotty.c	source to the 'dotty window' demo
dualplay.c	dual playfield example
flood.c	flood fill example
freemap.c	old version of 'freemap'
geltools.c	tools for VSprites and BOBs
gfxmem.c	graphic memory usage indicator
hello.c	window example from RKM
inputdev.c	adding an input handler to the input stream
joystik.c	reading the joystick
keybd.c	direct keyboard reading
layertes.c	layers examples
mousport.c	test mouse port
ownlib.c,	
ownlib.asm	example of making your own library with Lattice
paratest.c	tests parallel port commands
seritest.c	tests serial port commands
serisamp.c	example of serial port use
prinintr.c	sample printer interface code
prtbase.h	printer device definitions
regintes.c	region test program
setlace.c	source to interlace on/off program
setparallel.c	set the attributes of the parallel port
SetSerial.c	set the attributes of theserial port
singplay.c	single playfield example
speechtoy.c	source to narrator and phonetics demo
timedely.c	simple timer demo
timer.c	exec support timer functions
timrstuf.c	more exec support timer functions
WhichFont.c	loads and displays all available system fonts

process.i and prtbase.i assmebler include files:

autorqstr.txt	warnings of deadlocks with autorequesters
consoleIO.txt	copy of the RKM console I/O chapter
diskfont.txt	warning of disk font loading bug
fullfunc.txt	list of #defines, macros, functions
inputdev.txt	preliminary copy of the input device chapter

License information on Workbench distribution license

printer pre-release copy of the chapter on printer drivers, from RKM 1.1 v11fd.txt 'diff' of .fd file changes from version 1.0 to 1.1

v28v1.diff 'diff' of include file changes from version 28 to 1.0

AMICUS Disk 5 **Files from the Amiga Link / Amiga Information Network**

Note that some of these files are old, and refer to older versions of the operating system. These files are from Amiga Link. For a time, Commodore supported Amiga Link, aka AIN, for online developer technical support. It was only up and running for several weeks. These files do not carry a warranty, and are for educational purposes only. Of course, that's not to say they don't work.

A demo of Intuition menus called 'menudemo', In C source

whereis.c	find a file searching all subdirectories
bobtest.c	BOB programming example
sweep.c	sound synthesis example

Assembler files:

mydev.asm	sample device driver
mylib.asm	sample library example
mylib.i	
mydev.i	
asmsupp.i	
macros.i	assembler include files

Texts:

amigatricks	tips on CLI commands
extdisk	external disk specification
gameport	game port spec
parallel	parallel port spec
serial	serial port spec
v1.1update	list of new features in version 1.1
v1.1h.txt	'diff' of include file changes fr om V1.0 to 1.1

Files for building your own printer drivers, including dospecial.c, epsondata.c, init.asm, printer.c, printer.link, printertag.asm, render.c, and wait.asm. This disk does contain a number of files describing the IFF specification. These are not the latest and greatest files, but remain here for historical purposes. They include text files and C source examples. The latest IFF spec is elsewhere in this library.

AMICUS Disk 6 **IFF Pictures**

This disk includes the DPSlide program, which can view a given series of IFF pictures, and the 'showpic' program, which can view each file at the click of an icon. The pictures include a screen from ArticFox, a Degas dancer, the guys at Electronic Arts, a gorilla, horses, King Tut, a lighthouse, a screen from Marble Madness, the Bugs Bunny Martian, a still from an old movie, the Dire Straits moving company, a screen from Pinball Contruction Set, a TV newcaster, the PaintCan, a world map, a Porsche, a shuttle mission patch, a tyrannosaurus rex, a planet view, a VISA card, and a ten-speed.

AMICUS Disk 7 **DigiView HAM demo picture disk**

This disk has pictures from the DigiView hold-and-modify video digitizer. It includes the ladies with pencils and lollypops, the young girl, the bulldozer, the horse and buggy, the Byte cover, the dictionary page, the robot and Robert. This includes a program to view each picture separately, and all together as separate, slidable screens. The 'seeilbm' program, to turn any screen into an IFF picture.

AMICUS Disk 8

C programs:

Browse	view text files on a disk, using menus S-E-D
Crunch	removes comments and white space from C files, S-E
IconExec	EXECUTE a series of commands from Workbench S-E
PDScreen Dump	dumps Rastport of highest screen to printer
SetAlternate	sets a second image for an icon, when clicked once S-E
SetWindow	makes windows for a CLI program to run under Workbench S-E
SmallClock	a small digital clockin a window menu bar
Scrimper	the screen printer in the fourth AC S-E

Amiga Basic Programs:

(**Note**: Many of these programs are present on AMICUS Disk 1. Several of these were converted to Amiga Basic, and are included here.)

AddressBook	a simple address book database
Ball	draws a ball
Cload	program to convert Compuserve hex files to binary, S-D
Clue	the game, Intuition driven
ColorArt	art drawing program
DeluxeDraw	the drawing program in the 3rd AC, S-D
Eliza	conversational computer psychologist
Othello	the game, as known as 'go'
RatMaze	3D ratmaze game
ROR	boggling graphics demo
Shuttle	draws 3D pictures of the space shuttle
Spelling	simple spelling program
YoYo	wierd zero-gravity yo-yo demo, tracks yo-yo to the mouse

Executable programs:

3Dcube	Modula-2 demo of a rotating cube
AltIcon	sets a second icon image, displayed when the icon is clicked
AmigaSpell	a slow but simple spell checker, E-D
arc	the ARC file compression progra must-have for telecom, E-D
Bertrand	graphics demo
disksalvage	prog. to rescue trashed disks, E-D
KwikCopy	a quick but nasty disk copy program: ignores errors, E-D
LibDir	lists hunks in an object file E-D
SaveILBM	saves any screen as IFF pic.E-D ??
ScreenDump	shareware screen dump prog. E only
StarTerm	version 2.0, term program, XmodemE-D

Texts:

LatticeMain	tips on fixing _main.c in Lattice
GDiskDrive	make your own 5 1/4 drive
GuruMed	explains the Guru numbers
Lat3.03bugs	bug list of Lattice C version 3.03
MForgeRev	user's view of the MicroForge HD
PrintSpooler	EXECUTE-based print spool prog.

.BMAP files:

These are the necessary links between Amiga Basic and the system libraries. To take advantage of the Amiga's capabilities in Basic, you need these files. BMAPs are included for 'clist', 'console', 'diskfont', 'exec', 'icon', 'intuition', 'layers', 'mathffp', mathieeedoubas', 'mathieeesingbas', 'mathtrans', 'potgo', 'timer' and 'translator'.

AMICUS Disk 9

Amiga Basic Programs:

FlightSim	simple flight simulator program
HuePalette	explains Hue, Saturation, & Intensity
Requester	ex. of requesters from Amiga Basic
ScrollDemo	demonstrates scrolling capabilities
Synthesizer	sound program
WorldMap	draws a map of the world

Executable programs:

Boing!	latest Boing! demo,with selectable speed,E
Brush2C	converts an IFF brush to C data instructions, initialization code, E
Brush2Icon	converts IFF brush to an icon, E
Dazzle	graphics demo, tracks to mouse, E
DeciGEL	assembler program for stopping 68010 errors, S-E-D
Klock	menu-bar clock and date display, E
life	the game of life, E
TimeSet	Intuition-based way to set the time & date
EMEmacs	another Emacs, more oriented to word processing, S-E-D
MyCLI	a CLI shell, works without the Workbench, S-E-D

Texts:

FnctnKeys	read function keys from Amiga Basic
HackerSln	explains how to win the game 'hacker'
lst68010	guide to installing a 68010 in your Amiga
Boing!	latest Boing! demo,with selectablespeed, E
Brush2C	converts an IFF brush to C data instructions, initialization code, E
Brush2Icon	converts IFF brush to an icon, E
Dazzle	graphics demo, tracks to mouse, E
DeciGEL	assembler program for stopping 68010 errors, S-E-D
Klock	menu-bar clock and date display, E
life	the game of life, E
TimeSet	Intuition-based way to set the time date,
EMEmacs	another Emacs, more oriented to word processing, S-E-D
MyCLI	a CLI shell, works without the Workbench, S-E-D

For PDS orders, please use form on page 128

Texts:

FnctnKeys	explains how to read function keys from Amiga Basic
HackerSln	explains how to win the game 'hacker'
Ist68010	guide to installing a 68010 in your Amiga
PrinterTip	sending escape sequences to your printer
StartupTip	tips on setting up your startup-sequence file
XfrmrReview	list of Transformer programs that work

Printer Drivers:

Printer drivers for the Canon PJ-1080A, the C Itoh Prowriter, an improved Epson driver that eliminates streaking, the Epson LQ-800, the Gemini Star-10, the NEC 8025A, the Okidata ML-92, the Panasonic KX-P10xx family, and the Smith-Corona D300, with a document describing the installation process.

AMICUS Disk 10 **Instrument sound demos**

This is an icon-driven demo, circulated to many dealers. It includes the sounds of an acoustic guitar, an alarm, a banjo, a bass guitar, a boink, a calliope, a car horn, claves, water drip, electric guitar, a flute, a harp arpegio, a kickdrum, a marimba, a organ minor chord, people talking, pigs, a pipe organ, a Rhodes piano, a saxophone, a sitar, a snare drum, a steel drum, bells, a vibrophone, a violin, a wailing guitar, a horse whinny, and a whistle.

AMICUS Disk 11

C programs

dirutil	Intuition-based, CLI replacement manager S-E
cpri	shows and adjusts priority of CLI processes, S-E
ps	shows info on CLI processes, S-E
vidtex	displays Compuserve RLE pics, S-E

AmigaBasic programs

pointered	pointer and sprite editor program
optimize	optimization ex ample from AC article
calendar	large, animated calendar, diary and date book program
amortize	loan amortizations
brushtoBOB	converts small IFF brushes to AmigaBasic BOB OBJECTS
grids	draw and play waveforms
hilbert	draws Hilbert curves
madlib	mad lib story generator
mailtalk	talking mailing list program
meadows3D	3D graphics program, from A C™ article
mousetrack	mouse tracking example in hires mode
slot	slot machine game
tictactoe	the game
switch	pachinko-like game
weird	makes strange sounds

Executable programs

cp	unix-like copy command, E
cls	screen clear, S-E
diff	unix-like stream editor uses 'diff' output to fix files
pm	chart recorder performances indicator

Assembler programs

cls	screen clear and CLI arguments example

Modula-2

trails	moving-worm graphics demo
caseconvert	converts Modula-2 keywords to uppercase
Forth	Breshehan circle algorithm example
Analyze	12 templates for the spreadsheet Analyze

There are four programs here that read Commodore 64 picture files. They can translate Koala Pad, Doodle, Print Shop and News Room graphics to IFF format. Getting the files from your C-64 to your Amiga is the hard part.

AMICUS Disk 12

Executable programs

blink	'alink' compatible linker, but faster, E-D
clean	spins the disk for disk cleaners, E-D
epsonset	sends Epson setttings to PAR from menu E-D
showbig	view hi-res pics in low-res superbitmap, E-D
speaktime	tell the time, E-D
undelete	undeletes a file, E-D
cnvapldhm	converts Apple][low, medium and high res pictures to IFF, E-D
menued	menu editor produces C code for menus, E-D
quick	quick disk-to-disk nibble copier, E-D
quickEA	copies Electronic Arts disks, removes protection, E-D
txed 1.3	demo of text editor from Microsmiths,E-D

C programs

spin3	rotating blocks graphics demo, S-E-D
popcli	start a new CLI at the press of a button, like Sidekick, S-E-D
vsprite	VSprite example code from Commodore, S-E-D
AmigaBBS	Amiga Basic bulletin board prog., S-D

Assembler programs

star10	makes star fields like Star Trek intro,S-E-D

Pictures

Mount Mandelbrot	3D view of Mandelbrot set
Star Destroyer	hi-res Star Wars starship
Robot	robot arm grabbing a cylinder

Texts

vendors	Amiga vendors, names, addresses
cardco	fixes to early Cardco memory boards
cinclude	cross-reference to C include files
mindwalker	clues to playing the game well
slideshow	make your own slideshows from the Kaleidoscope disk

AMICUS Disk 13

Amiga Basic programs

Routines from Carolyn Scheppner of CBM Tech Support, to read and display IFF pictures from Amiga Basic. With documentation. Also included is a program to do screen prints in Amiga Basic, and the newest BMAP files, with a corrected ConvertFD program. With example pictures, and the SaveILBM screen capture program.

Routines to load and play FutureSound and IFF sound files from Amiga Basic, by John Foust for Applied Visions. With documentation and C and assembler source for writing your own libraries, and interfacing C to assembler in libraries. With example sound.

Executable programs

gravity	Sci Amer Jan 86 gravitation graphic simulation, S-E-D

Texts

MIDI	make your own MIDI instrument interface, with documentation and a hi-res schematic picture.

AMICUS Disk 14

Several programs from Amazing Computing issues:

Tools

Dan Kary's	C structure index program, S-E-D

Amiga Basic programs:

BMAP Reader	by Tim Jones
IFFBrush2BOB	by Mike Swinger
AutoRequester	example
DOSHelper	Windowed help system for CLI commands, S-E-D
PETrans	translates PET ASCII files to ASCII files, S-E-D
C Squared	Graphics program from Scientific American, Sept 86, S-E-D
crlf	adds or removes carriage returns from files, S-E-D
dpdecode	decrypts Deluxe Paint, remo
ves copy	protection, E-D
queryWB	asks Yes or No from the user returns exit code, S-E
vc	VisiCalc type spreadsheet, no mouse control, E-D
view	views text files with window and
slider	gadget, E-D

Oing, Sproing, yaBoing, Zoing are sprite-based

Boing! style demos, S-E-D

CLIClock, sClock, wClock are window border clocks, S-E-D

Texts

An article on long-persistance phospor monitors, tips on making brushes of odd shapes in Deluxe Paint, and recommendations on icon interfaces from Commodore-Amiga.

AMICUS Disk 15

The C programs include:

'pr'	a file printing utility, which can print files in the background, and with line numbers and control character filtering.
'fm'	displays a chart of the blocks allocated on a disk.
'Ask'	questions an 'execute' file, returns an error code to control the execution in that batch file
'Stat'	an enhanced version of AmigaDOS 'status' command.
'Dissolve'	random-dot dissolve demo displays IFF picture slowly, dot by dot, in a random fashion.
'PopCLI2'	invoke new CLI window at the press of a key.

The executable programs include:

'Form'	file formatting program through the printer driver to select print styles
'DiskCat'	catalogs disks, maintains, sorts,merges lists of disk files
'PSound'	SunRize Industries' sampled sound editor & recorder
'Iconmaker'	makes icons for most programs
'Fractals'	draws great fractal seascapes and mountain scapes.
'3D Breakout'	3D glasses, create breakout in a new dimension
'AmigaMonitor'	displays lists of open files,
memory use,	tasks, devices and ports in use.
'Cosmoroids'	version of 'asteroids' for the Amiga.
'Sizzlers'	high resolution graphics demo written in Modula 2.

Texts:

'ansi.txt'	explains escape sequences the CON: device responds to.
'FKey'	includes template for making paper to sit in the tray at the top of the Amiga keyboard.
'Spawn'	programmer's document from Commodore Amiga, describs ways to use the Amiga's multitasking capabilities in your own programs.

AmigaBasic programs:

'Grids'	draw sound waveforms, and hear them played.
'Light'	a version of the Tron light-cycle video game.
'MigaSol'	a game of solitaire.
'Stats'	program to calculate batting averages
'Money'	"try to grab all the bags of money that you can."

AMICUS 15 also includes two beautiful IFF pictures, of the enemy walkers from the ice planet in Star Wars, and a picture of a cheetah.

AMICUS Disk 16

'juggler' demo by Eric Graham. a robot juggler bouncing three mirrored balls, with sound effects. Twenty-four frames of HAM animation are flipped quickly to produce this image. You control the speed of the juggling. The author's documentation hints that this program might someday be available as a product.

IFF pictures

parodies of the covers of Amiga World and Amazing Computing magazines.

C programs:

'Inputhandler'	example of making an input handler.
'FileZap3'	binary file editing program
'ShowPrint'	displays IFF picture, and prints it.
'Gen'	program indexes and retrieves C structures and variables declared in the Amiga include file system.

Executable Programs:

'FixHunk2'	repairs an executable program file for expanded memory
'ms2smus'	converts Music Studio files to IFF standard "SMUS" format. I have heard this program might have a few bugs, especially in regards to very long songs, but it works in most cases.
'Missile'	Amiga version of the 'Missile Command' video game.

This disk also contains several files of scenarios for Amiga Flight Simulator II. By putting one of these seven files on a blank disk, and inserting it in the drive after performing a special command in this game, a number of interesting locations are preset into the Flight Simulator program. For example, one scenario places your plane on Alcatraz, while another puts you in Central Park

AMICUS Disk 17

Telcommunications disk whuch contains six terminal programs.

"Comm" V1.33	term prog. with Xmodem, WXmodem,
"ATerm" V7.2	term prog. includes Super Kermit
"VT-100"V2.6	Dave Wecker's VT-100 emulator with Xmodem,Kermit, and scripting
"Amiga Kermit"	V4D(060) port of the Unix C-Kermit
"VTek" V2.3.1	Tektronix graphics terminal emulator based on the VT-100 prog. V2.3 and contains latest 'arc' file compression
"AmigaHost"	V0.9 for Compuserve. Includes RLE graphics abilities & CIS-B file transfer protocol.
"FixHunk"	expansion memory necessity
"FixObj"	removes garbage characters from modem received files
"Txt"	filters text files from other systems to be read by the Amiga E.C.
"addmem"	executeable version for use with mem expansion article in AC v2..1
'arc'	file documentation and a basic tutorial on un 'arc'ing files
"arcre"	for makeing "arc" files E.C.

AMICUS Disk 18

Logo	Amiga version of the populacomputer language, with example programs, E-D
Tv*Text	Demo version of the TV*Text character generator
PageSetter	Freely distributable versions of the updated PagePrint and PageIFF programs for the PageSetter desktop publishing package.
FullWindow	Resizes any CLI window using only CLI commands, E-D
Life3d	3-D version of Conway's LIFE program, E-D
Defdisk	CLI utility to re-assign a new Workbench disk, S-E-D
Calendar.WKS	Lotus-compatible worksheet that makes calendars
SetKey	Demo of keyboard key re-programmer, with IFF picture to make function key labels, E-D
VPG	Video pattern generator for aligning monitors, E-D
HP-10C	Hewlett-Packard-like calculator, E-D
SetPrefs	Change the Preferences settings on the fly, in C, S-E-D
StarProbe	Program studies stellar evolution. C source included for Amiga and MS-DOS, S-E-D
ROT	C version of Colin French's AmigaBasic ROT program from Amazing Computing. ROT edits and displays polygons to create three dimensional objects. Up to 24 frames of animation can be created and displayed. E-D
Scat	Like Ing, windows on screen run away from the mouse, E-D
DK	Decays" the CLI window into dust, in Modula 2, S-E-D
DropShadow2	Adds layered shadows to Workbench windows, E-D

AMICUS Disk 19

This disk carries several programs from Amazing Computing. The IFF pictures on this disk include the Amiga Wake part T-shirt logo, a sixteen-color hi-res image of Andy Griffith, and five Amiga Live! pictures from the Amazing Stories episode that featured the Amiga.

Solve	Linear equation solver in assembly language, S-E-D
Gadgets	Bryan Catley's AmigaBasic Tutorial,
Household	Bryan Catley's AmigaBasic household inventory program, S-D
Waveform	Jim Shields' Waveform Wo[illegible]AmigaBasic, S-D
DiskLib	John Kennan's AmigaBasic disk librarian program, S-D
Subscripts	Ivan Smith's AmigaBasic subscript example, S-D
String, Boolean	C programs and executables for Harriet Maybeck Tolly's Intuition tutorials, S-E-D
Skinny C	Bob Riemersma's example for making small C programs, S-E-D
COMAL.h	Make C look like COMAL header file,
EmacsKey	Makes Emacs function key definitions by Greg Douglas, S-D
AMon 1.1	Snoop on system resource use, E-D
BTE	Bard's Tale character editor, E-D
Size	CLI program shows the size of a given set of files, E-D
WinSize	CLI window utility resizes current window, S-E-D

AMICUS Disk 20

Compactor, Decoder Steve Michel AmigaBasic tools, S-D

BobEd	BOB and sprite editor written in C,S-E-D
SpriteMasterII	Sprite editor and animator by Brad Kiefer, E-D
BlitLab	Blitter chip exploration C program by Tomas Rokicki, S-E-D
FPic	Image processing program by Bob Bush loads and saves IFF images, changes them with several techniques, E-D
Bankn	Complete home banking program, balance your checkbook! E-D

AMICUS Disk 21

Target	Makes each mouse click sound like a gunshot, S-E-D
Sand	Simple game of sand that follows the mouse pointer, E-D
PropGadget	Harriet Maybeck Tolly's proportional gadget example, S-E
EHB	Checks to see if you have extra-half-bright graphics, S-E-D
Piano	Simple piano sound program
CelScripts	Makes cel animation scripts for Aegis Animator, in AmigaBasic

This disk has electronic catalogs for AMICUS disks 1 to 20 and Fish disks 1 to 80. They are viewed with the DiskCat program, included here.

AMICUS Disk 22

Cycles	Light cycle game, E-D
Show_PrintIl	Views and prints IFF pictures, including larger than screen
PrtDrvGen2.3	Latest version of a printer driver generator
Animations	VideoScape animations of planes and boing ball
Garden	Makes fractal gardenscapes
BasicSorts	Examples of binary search and insertion sort in AmigaBasic

AMICUS Disk 23

An AMICUS disk completely dedicated to music on the Amiga. This disk contains two music players, songs, instruments, and players to bring the thrill of playing "Big Sound" on your Amiga

Instruments	a collection of 25 instruments for playing and creating music. The collection ranges from Cannon to Marimba
List INSTR	program to list the instruments DMCS will not load as well as list the origins for any instrument.
Music	a collection of 14 Classical pieces
1812Overture	The 16 minute classical feature complete with Cannon!

Three Amiga Music Players:

SMUSPlay
MusicCraft2SMUS
MusicStudio2SMUS

AMICUS Disk 24

Sectorama	A disk sector editor for any AmigaDOS file-structured device, recover files from a trashed hard disk. By David Joiner of MicroIllusions
Iconize	Reduces the size of IFF images, companion program, Recolor, remaps the palette colors of one picture to use the palette colors of another. Using these programs and a tool to convert IFF brushes to Workbench icons, make icons look like miniatures of the pictures.
CodeDemo	Modula-2 program converts assembler object files to inline CODE statements. Comes with a screen scrolling example
AmiBug	Workbench hack makes the same fly walk across the screen at random intervals. Otherwise, completely harmless.
BNTools	Three examples of assembly language code from Bryce Nesbitt: 1. SetLace,prog to switch interlace on&off. 2. Why, replace AmigaDOS CLI Why 3. LoadIt, prog to load a file into memory until a reboot. (Only the most esoteric hackers will find LoadIt useful.)
Monolace	CLI program resets Preferences to several colors of monochrome & interlace screens. C source is included, works with DisplayPref, a CLI program which displays the current Preferences settings.
BoingMachine	A ray-traced animation of a perpetual motion Boing-making machine, includes the latest version of the Movie program, which has the ability to play sounds along with the animation. By Ken Offer
Daisy	Example of using the translator and narrator devices to make the Amiga talk. It is written in C.
QuickFlix	Script-driven animation and slideshow program flips through IFF images.
BMon	System monitor AmigaBasic program ; perform simple manipulations of memory.
Moose	Random background program, a small window opens with a moose resembleing Bullwinklesaying witty phrases user definable.
DGCS	Deluxe Grocery Construction Set, simple Intuition-based prog for assembling and printing a grocery list.

The Virus Check directory holds several programs relating to the software virus that came to the US from pirates in Europe as detailed in Amazing Computing V2.12. Bill Koester's full explanation of the virus code is included. One program checks for the software virus on a Workbench disk; the second program checks for the virus in memory, which could infect other disks.

AMICUS Disk 25

Nemesis	Graphics demo pans through space towards the mythical dark twin of the sun with wonderful music and space graphics.

The KickPlay directory holds text that describes several patches to the Kickstart disk. For Amiga 1000 hackers who feel comfortable patching a disk in hexadecimal, KickPlay offers the chance to automatically do an ADDMEM for old expansion memory, as well as the ability to change the picture of the "Insert Workbench" hand. A program is also included for restoring the correct checksum of the Kickstart disk.

KeyBird	BASIC prog edits keymaps, adjust the Workbench keymaps or create your own.

8ColorWB	Modifies the Workbench so three bitplanes are used, icons can have eight colors, instead of four, eight-color icons are included. Public domain program "zapicon" or "brush2icon" converts eight-color IFF brushes to icons, to use Deluxe Paint to make icons for this new Workbench.
BrushIcon	Converts brushes to icons (bizarr docs).
Egraph	Graphing prog reads [x,y] values from a file and displays them on the screen, similar to the same-named Unix program.
Keep 1.1	Message-managing program for telecommunications, lets you save messages from an online transcript to another file, understands the message format of the national networks and several types of bulletin board software. Moves through the transcript and save messages
Kill.fastdir	Speed up directory access, it creates a small file in each directory on a disk which contains the information about the files, will also remove all the "fastdir" files from each directory. by CLImate's authors

The LaceWB program changes between interlace and non-interlace Workbench. Previously, you were forced to reboot after changing Preferences to an interlaced screen. This program flips between the normal and extended screen heights.

PW_Utility	A shareware utility for ProWrite users, changes margin settings and font types.
Guru	A CLI program, prints out probable causes for Guru mediations; C source included.
DiskWipe	Latest from Software Distillery, removes files from directories or disk drives, much faster than "delete."
Snow	AmigaBasic makes snowflake designs.
Mlist	Mailing list database.
Softballstats	Maintain softball statistics/ team records.
Dodge	Short Modula-2 program moves the Workbench screen around after a period of time, prevents monitor burn-in.

AMICUS Disk 26

Todor Fay's SoundScape module code from his Amazing Computing articles. The source to Echo, Chord, TX, and VU is included. The Lattice and Manx C source code is here, along with the executable modules.

ImageMaker	Interesting tool edits Image structures for C, loads & saves C code directly.
Claz2	Update of prog to convert IFF images to PostScript files for printing on laser printers
SDBackup	Hard disk backup prog with Lempel-Ziv compression to reduce the necessary number of disks.
TCB	Prints information about tasks and processes in the system; assembler source is included.
FunBut	Lets a function key act like a rapid series of left mouse button events.
DC	A handy program for people who use an Amiga 1020 5 1/4 inch drive as an AmigaDOS floppy. A Workbench program that sends a DiskChange signal to the operating system: instead of typing "diskchange df2:" over and over again, just click on the icon. C source included.
System config	File makes screen 80 columns wide of text in the Scribble! word processor.
Dick2Ram	2 programs to move the Scribble! spelling dictionary to and from the RAM disk.
Lexical	Analyzes a text file and gives the Gunning-Fog, Flesch, and Kincaid indices which measure readability.
HexDump	Modula-2 program to display memory locations in hexadecimal.
Tartan	AmigaBasic; design Tartan plaids.
DirMaster	Disk catalog program.
BMP	plays 8SVX sampled sounds in the background while something else is happening in the Amiga, as your Amiga is booting, for example.
ShowPt	CLI program changes your pointer to a given pointer.

AMICUS 26 also has a collection of mouse pointers, & Workbench program to display them

Fred Fish Public Domain Software

Fred Fish Disk 1:

amigademo	Graphical benchmark for comparing amigas.
amigaterm	simple communications program with Xmodem
balls	simulation of the "kinetic thingy" with balls on strings
colorful	Shows off use of hold-and-modify mode.
dhrystone	Dhrystone benchmark program.
dotty	Source to the "dotty window" demo on the Workbench disk.
freedraw	A small "paint" type program with lines, boxes, etc.
gad	John Draper's Gadget tutorial program
gfxmem	Graphical memory usage display prog.
halfbrite	demonstrates "Extra-Half-Brite" mode, if you have it
hello	simple window demo
latffp	accessing the Motorola Fast Floating Point library from C
palette	Sample prog. to design color palettes.
trackdisk	Demonstrates use of the trackdisk driver.
requesters	John Draper's requester tutorial and example program.
speech	Sample speech demo program. Stripped down "speechtoy".
speechtoy	Another speech demo program.

Fred Fish Disk 2:

alib	Object module librarian.
cc	Unix-like frontend for Lattice C compiler.
dbug	Macro based C debugging package. Machine independent.
make	Subset of Unix make command.
make2	Another make subset command.
microemacs	Small version of emacs editor, with macros, no extensions
portar	Portable file archiver.
xrf	DECUS C cross reference utility.

Fred Fish Disk 3:

gothic	Gothic font banner printer.
roff	A "roff" type text formatter.
ff	A very fast text formatter
cforth	A highly portable forth implementation. Lots of goodies.
xlisp	Xlisp 1.4, not working correctly.

Fred Fish Disk 4:

banner	Prints horizontal banner
bgrep	A Boyer-Moore grep-like utility
bison	CNU Unix replacement 'yacc', not working.
bm	Another Boyer-Moore grep-like utility
grep	DECUS grep
kermit	simple portable Kermit with no connect mode.
MyCLI	Replacement CLI for the Amiga. V. 1.0
mandel	A Mandelbrot set program, by Robert French and RJ Mical

Fred Fish Disk 5

cons	Console device demo program with supporting macro routines.
freemap	Creates a visual diagram of free memory
input.dev	sample input handler, t raps key or mouse events
joystick	Shows how to set up the gameport device as a joystick.
keyboard	demonstrates direct communications with the keyboard.
layers	Shows use of the layers library
mandelbrot	IFF Mandelbrot program
mouse	hooks up mouse to right joystick port
one.window	console window demo
parallel	Demonstrates access to the parallel port.
printer	opening and using the printer, does a screen dump, not working
print.support	Printer support routines, not working.
proctest	sample process creation code, not working
region	demos split drawing regions
samplefont	sample font with info on creating your own
serial	Demos the serial port
singlePlayfield	Creates 320 x 200 playfield
speechtoy	latest version of cute speech demo
speech.demo	simplified version of speechtoy, with IO requests
text.demo	displays available fonts
timer	demos timer.device use
trackdisk	demos trakcdisk driver

Fred Fish Disk 6:

compress	like Unix compress, a file squeezer
dadc	analog clock impersonator
microemacs	upgraded version of microemacs from disk 2
mult	removes multiple occuring lines in files
scales	demos using sound and audio functions
setparallel	Allows changing parallel port parameters
setserial	Allows changing serial port parameters.
sortc	quicksort based sort program, in C
stripc	Strips comments and extra whitespace from C source

Fred Fish Disk 7:

This disk contains the executables of the game Hack V 1.0.1.

Fred Fish Disk 8:

This disk contains the C source to Hack on disk 7.

Fred Fish Disk 9:

moire	Draws moire patterns in black and white
MVP-FORTH	Mountain View Press Forth, version 1.00.03A. A shareware version of FORTH from Fantasia Systems.
proff	a more powerful text formatting program
setlace	Prog to toggle interlace mode on and off
skewb	a rubic's cube type demo
sparks	moving snake Graphics demo

Fred Fish Disk 10:

conquest	An interstellar adventure simulation game
dehex	convert a hex file to binary
filezap	Patch program for any type of file.
fixobj	Strip garbage off Xmodem transferred files.
iff	Routines to read and write iff format files.
ld	simple directory program
ls	Minimal UNIX ls, with Unix-style wildcarding, in C
sq,usq	file squeeze and unsqueeze
trek73	Star Trek game
yachtc	Dice game.

Fred Fish Disk 11:

dpslide	slide show program for displaying IFF images with miscellaneous pictures

Fred Fish Disk 12:

amiga3d	Shows a rotating 3 dimensional solid "Amiga sign".
ArgoTerm	a terminal emulator program, written in assembler
arrow3d	Shows a rotating 3 dimensional wire frame arrow.
ld4	directory listing program
IconExec	
SetWindow	two progs for launching progs from Workbench, presently only works under CLI.
SetAlternate	Makes an icon show a second image when clicked once
StarTerm	terminal emulator, with ASCII Xmodem, dialer, more.

Fred Fish Disk 13:

A Bundle of Basic programs, including:

Jpad	toybox	ezspeak	mandlebrot
xmodem	3dsolids	addbook	algebra
ror	amgseq1	amiga-copy	band
bounce	box	brickout	canvas
cardfi	circle	colorcircles	Copy
cubes1	cutpaste	datedogstar	
dragon	draw	dynamictriangle	
Eliza	ezterm	fillibuster	fractal
fscape	gomoku	dart	haiku
hal9000	halley	hauntedM	hidden
join	loz	mandel	menu
minipaint	mouse	Orthello	patch
pena	pinwheel	gboxrandom-circles	
Readme	rgb	rgbtest	Rord
sabotage	salestalk	shades	shapes
shuttle			
sketchpad	spaceart	speakspeach	
speecheasy	spell	sphere	
spiral	striper	superpad	suprshr
talk	terminal		
termtest	tom	topography	triangle
wheels	xenos	xmostriper	

(note: some programs are Abasic, most are Amigabasic, and some programs are presented in both languages)

Fred Fish Disk 14:

amiga3d	update of #12, includes C source to a full hidden surface removal and 3D graphics
beep	Source for a function that generates a beep sound
dex	extracts text from within C source files
dimensions	demonstrates N dimensional graphics
filezap	update of disk 10, a file patch utility
gfxmem	update of disk 1, graphic memory usage indicator
gi	converts IFF brush files to Image struct, in C text.
pdterm	simple ANSI VT100 terminal emulator, in 80 x 25 screen
shell	simple Unix 'csh' style shell
termcap	mostly Unix compatible 'termcap' implementation.

Fred Fish Disk 15:

Blobs	graphics demo, like Unix 'worms'
Clock	simple digital clock program for the title bar
Dazzle	An eight-fold symmetry dazzler program. Really pretty!
Fish	double buffered sequence cycle animation of a fish
Monopoly	A really nice monopoly game written in AbasiC.
OkidataDump	Okidata ML92 driver and WorkBench screen dump program.
Polydraw	A drawing program written in AbasiC.
Polyfractals	A fractal program written in AbasiC.

Fred Fish Disk 16:

A complete copy of the latest developer IFF disk

Fred Fish Disk 17:

The NewTek Digi-View video digitizer HAM demo disk

Fred Fish Disk 18:

AmigaDisplay	dumb terminal program with bell, selectable fonts
Ash	Prerelease C Shell-like shell program, history, loops, etc.
Browser	wanders a file tree, displays files, all with the mouse
MC68010	docs on upgrading your Amiga to use a MC68010
Multidim	rotate an N dimensional cube with a joystick
PigLatin	SAY command that talks in Pig Latin
Scrimper	Screen image printer
Xlisp1.6	source, docs, and execut for a Lisp interpret.

Fred Fish Disk 19:

BlackJack	text-oriented blackjack game
JayMinerSlides	Slides by Jay Miner, Amiga graphics chip designer, showing flowchart of the Amiga internals, in 640 x 400.
Keymap_Test	test program to test the key mapping routines
LockMon	Find unclosed file locks, for programs that don't clean up.

Fred Fish Disk 20:

AmigaToAtari	converts Amiga object code to Atari form
DiskSalv	program to recover files from a trashed AmigaDOS disk.
Hash	example of the AmigaDOS disk hashing function
Hd	Hex dump utility ala Computer Language magazine, April 86
MandelBrots	Mandelbrot contest winners
MultiTasking	Tutorial and examples for Exec level multitasking
Pack	strips whitespace from C source
PortHandler	sample Port-Handler program that performs. Shows BCPL environment
Random	Random number generator in assembly, f or C or assembler.
SetMouse2	sets the mouse port to right or left
SpeechTerm	terminal Emulator with speech capabilities, XModem
TxEd	Demo editor from Microsmith's Charlie Heath

Fred Fish Disk 21

This is a copy of Thomas Wilcox's Mandelbrot Set Explorer disk. Very good!

Fred Fish Disk 22

This disk contains two new "strains" of microemacs.

Lemacs	version 3.6 by Daniel Lawrence. For Unix V7, BSD 4.2, Amiga, MS-DOS, VMS. Uses Amiga function keys, status line, execute, startup files, more.
Pemacs	By Andy Poggio. New features include <ALT> keys as Meta keys, mouse support, higher priority, backup files, word wrap, function keys.

Fred Fish Disk 23

Disk of source for MicroEmacs, several versions for most popular operating systems on micros and mainframes. For people who want to port MicroEmacs to their favorite machine.

Fred Fish Disk 24:

Conques	interstaller adventure simulation game
Csh	update to shell on Disk 14, with built in commands,named variables substitution.
Modula-2	A pre-release version of the single pass Modula-2 compiler originally developed for MacIntosh at ETHZ. This code was transmitted to the AMIGA and is executed on the AMIGA with a special loader. Binary only.

Fred Fish Disk 25

Graphic Hack	A graphic version of the game on disks 7 and 8 This is the graphics-oriented Hack game by John Toebes. Only the executable is present.

Fred Fish Disk 26

UnHunk	Processes the Amiga "hunk" loadfiles. Collect code, data, and bss hunks together, allows individual specificatio of code, data, and bss origins, and generates binary file with format reminiscent of Unix "a.out" format. The output file can be easily processed by a separate program to produce Motorola "S-records" suitable for downloading to PROM programmer. By Eric Black.
C-kermit	Port of the Kermit file transfer program and server.
Ps	Display and set process priorities
Archx	Yet another program for bundling up text files and mailing or posting them as a single file unit.

Fred Fish Disk 27

ABdemos	Amiga Basic demos; Caroly Scheppner.
NewConvertFD	creates .bmaps from fd files.
BitPlanes	finds addresses of and writes to bitplanes of the screen's bitmap.
AboutBmaps	A tutorial on creation and use of bmaps.
LoadILBM	loads and displays IFF ILBM pics.
LoadACBM	loads and displays ACBM pics.
ScreenPrint	creates a demo screen and dumps it to a graphic printer.
Disassem	Simple 68000 disassembler. Reads standard Amiga object files and disassembles the code sections. Data sections are dumped in hex. The actual disassember routines are set up to be callable from a user prog so instructions in memory can be disassembled dynamically. By Bill Rogers.
DvorakKeymap	Example of a keymap structure for the Dvorak keyboard layout. Untested but included because assembly examples are few and far between. By Robert Burns
Hypocycloids	Spirograph, from Feb. 84 Byte.
LinesDemo	Example of proportional gadgets to scroll a SuperBitMap.
MemExpansion	Schematics and directions for building your own homebrew 1 Mb memory expansion, by Michael Fellinger.
SafeMalloc	Program to debug 'malloc()' calls
ScienceDemos	Convert Julian to solar and sidereal time, stellar positions and radial velocity epoch calculations and Galilean satellite plotter. By David Eagle.

Fred Fish Disk 28

ABasic games by David Addison: Backgammon , Cribbage, Milestone, and Othello

Cpp	DECUS 'cpp' C preprocessor, & a modified 'cc' that knows about the 'cpp', for Manx C.
Shar	Unix-compatible shell archiver, for packing files for travel.
SuperBitMap	Example of using a ScrollLayer, syncing SuperBitMaps for printing, and creating dummy RastPorts.

Fred Fish Disk 29

AegisDraw Demo	Demo program without save and no docs.
Animator Demo	Player for the Aegis Animator files
Cc	Unix-like front-end for Manx C
Enough	Tests for existence of system resources, files, and devices
Rubik	Animated Rubik's cube program
StringLib	
Vt100	VT-100 terminal emulator with Kermit and Xmodem protocols

Fred Fish Disk 30

Several shareware programs. The authors request a donation if you find their program useful, so they can write more software.

BBS	an Amiga Basic BBS by Ewan Grantham
FineArt	Amiga art
FontEditor	edit fonts, by Tim Robinson
MenuEditor	Create menus, save them as C source, by David Pehrson
StarTerm3.0	Very nice telecom by J. Nangano

(Fred Fish Disk#30 is free if requested when ordered with at least three other disks from the collection.)

Fred Fish Disk 31

Life	Life game, uses blitter to do 19.8 generations a second.
Mandelbrot	Version 3.0 of Robert French's program.
MxExample	Mutual exclusion gadget example.
RamSpeed	Measure relative RAM speed, chip and fast.
Set	Replacement for the Manx "set" command for environment variables,with improvements.
Tree	Draws a recursive tree, green leafy type, not files.
TxEd	Crippled demo version of Microsmith's text editor, TxEd.
VDraw	Full-featured drawing program by Stephen Vermeulen.
Xicon	Invokes CLI scripts from icon
Ticon	Displays text files from an icon.

Fred Fish Disk 66

AmScsi	Preliminary plans for a SCSI disk controller board.
Asm68k	Macro assembler, version 1.0.1. E-D
Assigned	Example for avoiding DOS insert-disk requester, by scanning the list of 'assign'ed names. S-E-D
Dk	Pretends to eat away at CLI window. S-E-D
Flip	Flips whole screen as a joke. S-E-D
Foogol	Foogol cross-compiler generates VAX assembly code. S-E-D
Free	Prints amount of free space on all drives.S-E-D
MallocTest	malloc/free memory test program. S-E-D
Melt	Pretends to melt the screen. S-E-D
Nart	Graphic flying string demo. S-E-D
Purty	Easy way to set printer attributes from Workbench. E-D
RayTracer	Simple ray tracing program. E-D
SendPackets	Updated CBM examples of packet routines on disk 35. S-E-D
SnapShot	Memory resident screen dump. E-D
TagBBS	Shareware BBS system, version 1.02.

Fred Fish Disk 67

AmCat	Shareware disk cataloging program.
AmigaSpell	Shareware Intuition spelling checker, V2.0. E-D
Bouncer	3-D bouncing ball written in MultiForth, SED
Comm	Terminal program version 1.33, E
Dux5	Another version of DirUtil. S-E-D
HexCalc	Hex, octal, & decimal calculator. E-D
Icons	Various big and alternate image icons.
Mandala	Mandala graphics and sound. E
PersMait	Demo shareware personal file manager.
RSLClock	Menu bar clock version 1.3. E-D
RTCubes	Graphics demo of 3D cubes. E-D
Wheel	"Wheel of Fortune"-type game,AmigaBASIC

Fred Fish Disk 68

This is version MG 1b of the MicroGNUEmacs. Source and executable areincluded, as well as source for other computers besides the Amiga.

Fred Fish Disk 69

Asm68k	Macro assembler, v1.0.3, E-D
BlitLab	Blitter exploring program, in C, S-E-D
Conman	Replacement console device handler adds editing and history to any application that uses CON:, v0.9, E-D
Console	Replacement console routines, in C, S-E-D
Dk	Decays the screen bit by bit, update to disk 66, in Modula-2, S-E-D
Frags	Displays memory fragmentationby listing the size of free memory blocks, in C, S-E-D
IconType	Change the type of an icon, in C, S-E-D
Make	'make' in Manx C, S-E-D
MonProc	Monitors processes for packet activity, in C,S-ED
MouseClock	Mouse pointer into a digital clock,in C,SED
Sb	Browses system structures, from Transactor magazine, v1.0, in C, S-E-D
Spew	Generates 'National Enquirer'-type headlines from rules file. In C,S-E-D
Spool	Three programs to demonstrate multitasking & spooling in a printer spooler. In C, v1.2, S-E-D
Wc	Counts words ala Unix 'wc', but faster, in C,S-ED

Fred Fish Disk 70

This is a disk of shareware programs.

AmigaMonitor	Explores state of the system, v1.13
Arc	Standard file compressor andl librarian, v0.23, a port of MS-DOS v5.0. E-D
BlackBook	Phone book program.
DoTil	Intuition-driven file manipulator program,v2.0.
GravityWars	Game of planets, ships and black holes,v1.03.
Jobs	Alternate user interface to CLI and WB, v2.1.
Lens	Magnifies area around mouse, shows it in a window, v1.0.
Life-3d	3D version of the classic cellular-automaton game, v1.2.
Logo	Logo language interpreter
SetKey	Demo keymap editor, v1.0
Vpg	Makes displays for aligning video monitors, v1.0.

Fred Fish Disk 71

AirFoil	Makes airfoils using the Joukowski transformation, in C, S-E-D
Amiga Basic	Miscellaneous programs including 3D plot program, a kaleidoscope, C-A logo drawing program file comparison utility string search program, S-E-D
Blocks	A variation of "lines", but with variable color blocks. E-D
Comm	Great terminal program, v1.34, E-D
DiskX	Utility for exploring file system.E-D
Fpic	Simple image processing program that operates on IFF pictures, with several filters, merging images, E-D
IconMk	Makes icons for files , v1.2a, E-D
Icons	New icons
NewFonts	Two new fonts; 'shalt18', an electronic ciruit element font, and 'ibm5', a PC-like font.
PetCLI	An AmigaBASIC CLI shell program.
PWDemo	Demo of the commercial product PowerWindows,v1.2. It aids creation of custom windows, menus, and gadgets, giving C or assembly source. E-D
Rot	Creates and animates 3-D objects, v0.5, E,D
TimeSet	Sets time from Workbench, E-D

Fred Fish Disk 72

This is a disk of IFF pictures.

Fred Fish Disk 73

Add	Customizes existing program menus with Amiga-key shortcuts. Also includes 'until', which waits until a given window is created. Shareware, in C, S-E-D.
AutoIconOpen	Fools WB into thinking mouse has double-clicked icons. In C, S-E-D
Dio	Generic Exec device interface code for opening libraries, getting multiple I/O channels, asynchronous operations, etc. In C, S-E-D.
Dissolve	Slowly displays IFF files, ala Nov 86 Dr. Dobb's program. In C, S-E-D
DTerm	Flexible, reprogrammable terminal program v1.10, E-D
Expose	Re-arranges windows so that at least one pixel of menu bar gadgets are exposed. InC, S-E-D.
Lit	Scans a text file, converts to C-style printable strings.C,v2.0, S-E-D
Lmv	"Long Movie", program views series of IFF picts in quick succession, upto 19 fps. Shareware, E-D
MouseOff	Mouse pointer disappears after ten seconds of non-use. In C, S-E-D
ParOut	Examples of controlling parallel port with resources instead of the PAR: device.In C,S-E-D
PenPalFont	Child-like font.
RunBackGround	Similar to RunBack on disk 66, runs program from the CLI allowing the CLI window to close. In C,S-E-D
SnapShot	Screendump utility,update FF 66.E-D
TypeAndTell	Example installs a device handler before Intuition, and speaks each key as it is pressed. In C and assembler, S-E-D
Xplor	Prints info about system lists, in assembler,S-E-D

Fred Fish Disk 74

Cled	Edits and recalls CLI commands, v1.3, E-D
Control	Intercepts graphic printer dump calls and accesses color map, width,and screen resolution. C,S-E-D
Dme	Simple WYSIWYG text editor for programmers,v1.25. Update of FF 59.E-D
DropShadow	WB dropshadows, v2.0.UpdateFF59. E-D
Funds	AmigaBASIC prog tracks mutual or stocks-D
Less	Text viewing program, like Unix "more", v1.1, update to disk 34. S-E-D
Makemake	Scans C source files and constructs a vanilla 'makefile' in the current directory. S-E-D
mCAD	Object-oriented drawing prog. v1.2.4, update to FF 59.Shareware, E-D
Random	Simple random number generator in C. S-E-D
TDebug	Monitors devices by intercepting Exec SendIO() and DoIO() vectors, in C, v1.0, S-E-D
Units	Converts measurements in different units, includes "chart" option, in C, S-E-D
XCopy	Replacement for AmigaDOS 'copy', doesn't change the date, uses Unix wildcards. E-D

Fred Fish Disk 75

Bezier	Play with Bezier curves points and granularity, S-E-D
BSplines	Play with b-splines, as above, S-E-D
Comm	C source for Comm terminal program v1.34. S-E-D
Copy	Replacement 'copy' command v1.0, preserves date, in C, S-E-D
Diff	Simple 'diff' in C, S-E-D
DuM2	Another DirUtil in Modula-2, v1.5, S-E-D
Eless	Fast 'dir' program in C, S-E-D
Fd	Faster 'eless' in C, S-E-D
HardCopy	Sends a transcript of a CLI session to a file, in C, S-E-D
MouseOff	Update FF73, turns off mouse pointer, S-E-D
SetFont	Changes the font in a Workbench screen, v2.0, S-E-D
SpeedDir	Another fast 'dir', in assembler, S-E-D

Fred Fish Disk 76 & 77

These are disks 1 and 2 of Chris Gray's Draco distribution for the Amiga. Draco is a compiled, structured language reminiscent of both C and Pascal. A full interface to AmigaDOS and Intuition is supplied. Be sure to get both disk 76 and 77.

Fred Fish Disk 78

Cycles	Cycle game like "Tron", v1.0, E-D
EOMS	Experts Only Mercenary Simulator game, E-D
MandelVroom	Mandelbrot generator with enhanced palette controls, fixed/floating point, presets, v1.50, in Manx C, S-E-D

Fred Fish Disk 79

AsmTools	CLI tools in assembler: echo, loadit, mounted, setlace, why; S-E-D
AssignDev	Give devices multiple names, in C, S-E-D
AuxHandler	Example of a dos handler that allows use of a CLI via the serial port. Includes source. Author: Steve Drew
Cmd	Redirects printer output to a file, in C, S-E-D
Info	AmigaDOS 'info' replacement, in C and assembler, S-E-D
Kill	Removes a task and its resources, in C,S-E-D
M2Error	Displays errors from TDI Modula-2 compiles, S-E-D
MonProc	Update to process packet prog. from FF69,in C,S,E,D
Mounted	Program for testing if a drive is present, in a script In C, S-E-D
Nro	Another 'roff'-style text formatter, in C, S-E-D
ParTask	Finds parent task, in C, S-E-D
QueryAny	For scripts, asks a question, accepts Y/N, gives return code. In assembler, S-E-D
ScnSizer	Resets pref settings for screen size, in C,SED
SharedLib	Example, shared lib, in C & assembler, S-E-D
Task	Simple CreateTask() example in C, S-E-D
Uw	Unix Windows client v1.0, in C, S-E-D
Who	Lists tasks on ready and wait queues, in C, S-E-D

Fred Fish Disk 80 (see Fred Fish 90)

Fred Fish 80 has been withdrawn due to copyright problems.

Fred Fish Disk 81

Asm68k	V1.1.0 of a macro assembler
AutoFacc	Shrinks the FACC window and moves it to the back
Brushes	53 custom IFF brushes of electronic symbols
CheckIFF	Checks structure of an IFF file CledV1.4 update FF74 of a simple CLI
Conman	Replaces console handler to add editing and history to many programs
Fonts	Miscellaneous fonts
Icon	V6.0 of the Icon programming language
KeyLock	Freezes the keyboard and mouse until pass word entered.
ScatDisplay	hack created from "lng"
Smush	Smushes an IFF file.
Target	Each mouse clikc becomes a gunshot

Fred Fish Disk 82

Adventure	Port of the classic Crowther and Woods game
AmicTerm	V0.50 of a telecommunications program, with scripts, redial, beeps, enhanced file requester
D2D-Demo	Demo version of Disk-2-Disk by Central Coast Software
DX-Synth	Voice filer program for Yamaha DX series synthesizers, update to disk 38
DiskMan	V1.0 of another DirUtil program
Icons	Miscellaneous new icons
Panl	Universal MIDI patch panel, v1.2
Rocket	Another Workbench hack, plays Lunar Lander
Sand	Game of sands following your pointer.

Fred Fish Disk 83

This disk contains a demo version of TeX from N Squared. It is limited to small files, and the previewer can only display ten pages or less, and only a small number of fonts are provided.

Fred Fish Disk 84

AudioTools	Programs from Rob Peck's July/August Amiga World article
BlitLab	Blitter experimentation program, V1.2, update to FF69
Ed	Simple editor, similar to Unix 'ed', based on the editor in Software Tools.
GravityWars	Game of planets, ships and black holes, v1.04, update to disk 70.
HunkPad	Adds legal padding to executables for Xmodem transmission.
PipeHandler	An AmigaDOS pipe device which supports named pipes and taps. V1.2
PopCLI	V3.0 of a hot-key to invoke a CLI window, with screen blanker, update to disk 40.
Requester	Update FF34, file requester similar to DPaint.
ScottDevice	V33.1 of a 'mount'able MicroForge SCSI driver.
Viacom	Another Schwab hack, makes TV-like static on screen.Parody

Fred Fish Disk 85

Csh	V2.06 of Dillon's 'csh'-like shell
FileReq	Source to wildcard file requester
Hide	Hides expansion memory from programs
ImageTools	Shareware tools to manipulation IFF images
LowMem	ServerShared library to aid in low memory situations
Plot6	A star plotting program with source.
RawIO	Example of setting raw mode on standard input
Rocket	Lunary Lander for Workbench, with source.
VMore	"more"-like text viewing utility, v1.0 SE
Vnews	Simple Unix news reader.

Fred Fish Disk 86

AutoPoint	Auto-selects window under the mouse pointer, with screensaver.
ClickToFront	Double-clicks in window brings it to front, v1.1, S-E-D
Cmd	V3.0 of a tool to redirect printer output toa file.
FileIISG-Demo	Demo of Softwood File Ilsg, a database manager with sound and graphics.

Fred Fish Disk 87

AdvSys	Adventure system from Byte May 1987, v1.2 E-D
AutoIconOpen	Fools Workbench to open disk icons, V1.2 update to disk 73, S-E-D
Claz	Converts IFF files to PostScript, V2.0, SED
Commodi	tiesMackraz's Commodities Exchange, an exec library to manage input handler, v0.4
Diff	Update to disk 75 of Unix-like 'diff', S-E-D
Dme	V1.27 of Dillon's text editor, update FF74,E-D
DropShadow	V2.0 of prog. that puts shadows on Workbench, S-E-D
Elib	Shared library example in Manx C.
ID-Handler	An AmigaDOS device handler generates unique identifiers, V1.0, S-E-D
Install	Alternate AmigaDOS 'install' programs,SED
MemWatch	Waits for low memory trashing, V2.0, SED
MovePointer	Moves pointer to given location, S-E-D
MoveWindow	Move window to given location, S-E-D
MunchingSq	Munching Squares hack, S-E-D
PalTest	Test to see if this is a PAL machine, S-E-D
Sc	Generates random scenery, S-E-D
Tek4695	Tek4695 printer driver
WBDualPF	Example of dual-playfield screen, update FF41, S-E-D
WarpText	Fast text rendering routines, S-E-D
Yaiff	rExample IFF reader, S-E-D
Zoo	A file archiver like "arc", v1.42A, E-D

Fred Fish Disk 88 (see Fred Fish 89)

FF Disk 88 has been removed due to copyright problems

Fred Fish Disk 89 (replaces Fred Fish 80)

DirMaster	Disk catalogue program, V1.0a, E-D
FuncKey	Shareware function key editor, V1.01, E-D
MFF-Demo	Demo of MicroFiche Filer database prog
ScreenShift	Adjust screen position ie Preferences,SED
Snake	Bouncing squiggly lines demo, S-E-D
AutoEnguirer	screen contraption requester improvement S-E-D
DemoLition	Display Hack S-E-D

Fred Fish Disk 90 (replaces Fred Fish 80)

AmiGazer	Night sky viewer of 1573 stars, set date, time, day. E-D
CardFile	AmigaBasic card file study aid. E-D
Conman	Console handler replacement gives line editing and history to most progs, v0.98,ED
IMandelVroom	Slight update to disk 78 Mandelbrot program, E-D
NewDemos	Replacements for lines and boxes demos that take less CPU time, E-D
Othello	Game of Othello, E-D
PrinText	Displays text files with gadgets, speech, IFF display, v1.2, E-D
PrtDrvGen	Automatic printer driv. generator,v2.2b,ED
RainBench	Cycles colors of WB backdrop or text. ED
ShortCut	Makes single-key shortcuts for entering commonly typed CLI commands,&custom macros. E-D
ShowPrint	Displays and prints all sizes of IFF pictures & controls printer output styles, v2.0 E-D
Sizzlers	Graphics demos, v1.7.0, E-D
Timer	Small Workbench timer counts time and $/ minute, E-D

Fred Fish Disk 91

Adventure Definition Language (ADL) a superset of an older language called DDL by Michael Urban, Chris Kostanick, Michael Stein, Bruce Adler, and Warren Usui. ADL enhancements by Ross Cunniff. Included are sources to the ADL compiler, interpreter, and debugger. Binaries combined by Ross with Lattice 3.03. CLI enviornment only. Documentation is available from the authors.

Fred Fish Disk 92

As6502	portable 6502 assembler, C source, by J. Van Ornum, Amiga port by Joel Swank
Bawk	Text processor update from FF65 Inspired by UNIX awk. Searches files for patterns, performs actions based on patterns. By Bob Brodt; Amiga port by Johan Widen
HunkPad	update of FF84 version. by J. Hamilton,pads an object file to a multiple of 128 bytes for better xmodem transfer. S E
Less	Like Unix "more", better, version 1.2 update of FF74. Scrolls Back and f orward. S E by Mark Nudelman, Amiga port by Bob Leivian.
Ndir	Library that implements the 4BSD unix dir access routines by MikeMeyer. S
Parse	Recursive descent expression parcer, computes, and prints expressions. includes transcendental function support. c Source included. by J Olsen
Shar	Two programs to pack and unpack shell archives includes C source. by Fabbian G. Dufoe
SmallLib	8 times smaller Amiga.lib replacement, binary only. by Bryce Nesbitt
UUencode	Encode/decode binary files for e mail or text-only methods. Update of FF53, includes checksum technique, compatible with older versions, plus transparent to older versions options. By Mark Horton, modified by Alan Rosenthal & Bryce Nesbitt.

Fred Fish Disk 93

Dme	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scro-lling, title-line statistics, multiple windows, ability to iconify windows. Update of FF87, SE. by Matt Dillon
MicroEmacs	Version 3.8, update to FF61 includes source. Orig by Dave Conroy modifications by Daniel Lawrence

Fred Fish Disk 94

AudioTools	Demo programs from Rob Peck's July/Augustt issue of AmigaWorld on accessing the audio device. V2,update of FF84. S. by Rob Peck
ClickUpFront	Similar in function to ClickToFront prog (FF86), bring windows tofront by clicking on any part of them. V 1.0. by Davide Cervone SE
HeliosMouse	Automatically activate a window simply by moving the mouse pointer into the window. V 1.0. Includes source. By Davide Cervone
IFF2Ps	Convert any IFF file to postscript for printing or viewing on a postscript compatible device. Version 1.2, by William Mason and Sam Paolucci E
ModulaTools	Various Modula 2 prog. routines. by Jerry Mack
Terrain3d	Pseudo-random 3d relief scenery generator, update of "sc", FF87. by Chris Gray, 3d by Howard Hull

Fred Fish Disk 95

Cmd	redirects the serial.device or parallel. device output to a file. Capture print jobs, debug or "offline" printing.V4 By C Scheppner SE
CygnusEdDemo	Demo of CygnusSoft's CygnusEd editor, a multiplefile, multiple feature editor.Includes demo 3.0 of MandFXP. by CygnusSoft Software E
Gomf	"Get Outa My Face" makes the Guru go away to allow clean-up & shutdown more cleanly. V 1.0, by Christian Johnsen E
Journal	records sequence of mouse & keyboard events, stored in a file for future playback. Good for demos or documtenting bugs. E. by D. Cervone
MergeMem	attempts merging of MemList entries of sequentially configured ram boards. Allows allocating a section of memory which spans both boards. V 2, update of FF56. by Carolyn Scheppner SE
PrinterStealer	Asimilar to "Cmd", allows diversion of output destined for printer to a file. Binary only, Source avai. from authors. by A. Livshits & J-M Forgeas
Record-Replay	similar to "Journal", records and plays back mouse and keyboard events. B only. source avail. from authors, Alex Livshits & J-M Forgeas

Fred Fish Disk 96

AnimPlayer	Animation reader and displayer by the combined efforts of Videoscape, Sculpt3D, Silver, Forms- In-Flight, and Animator:Appreticeby M Hashetal.
Chess	Amiga port, non-Amiga interface. High playability. V 1.0, S. by J. Stanback, Amiga port by B. Leivian
Hackbench	provides source for WB-Likeprog, for experimentation & validation of new interface ideas. Not a WB replacement. by Bill Kinnersley
Label	Print labels with arbitray text. V 1.3, Source available from author, M.Hansen
LineDrawer	Produces line drawings based on drawing commands stored in a text file. Includes demo that draws an outline map of the USA and state borders. V1.0, SE. by John Olsen
PopUpMenu	Example code implementing pop-up menus, reasonably compatible with Intuition menus.SE. by Derek Zahn
Tek4695	Tektronix 4695/4696 printer driver. SE. by P Staub
TimeRam	Fast and Chip ram test prog,E by B Takahashi
WarpText	Fast text rendering routines, to be linked with application progs.Text display "as fast or faster than 'blitz'". V2.0 update of FF87, S by Bill Kelly

Fred Fish Disk 97

Replaces FF57 for Copywrite problems

CutAndPaste	Implementaions of Unix cut and paste commands. by John Weald
GraphIt	Progran to plot simple functions in 2 or 3 dimensions. by Flynn Fishman
Juggler	V1.2 of robot juggler animation. Uses HAM mode and ray tracing. by Eric Graham
MouseReader	Shareware program to read text files & view IFF files using only the mouse. by William Betz
Splines	Prog to demonstrate curve fitting & ren-dering techniques. by Helene (Lee) Taran
Shm	Graphics demo, approximately simulates the motion of two interacting pendulums. Includes S by Chris Ediss

Notice: Fred Fish Disks 32 to 65 could not be listed in this issue. Please see our listing last issue for these disks.

Fred Fish Disk 98

Access	16 color terminal program based on Comm V1.,34. Includes Macro window, custom gadgets, colorized menus, etc. V. Beta 0.18 by Keith Young ,comm by D.J.James. E.
Backup	Writes AmigaDos disks as the backup destin-ation. recover files from the backup disk. Requires manual decisions on disk structure. by Alan Kent SE
DCDemo	DishCat 2.3, a disk catalog program, demo limited to cataloging 100 files at a time. by Ed Alford, MicroAce Software
HdDriver	WD-1002-05 hard disk controller driver. Card capable of maintaining 3 hard disks and 4 floppies, the driver is capable of only one hard disk. by Alan Kent SED
QBase	Quick-Base, a "MailBase Management utility", define and maintain a maximum of 200 records per file. by Kevin Harrise E
Thai	Thai language quiz program. Speak or type english/Thai sentences from supplied file. by Alan Kent SE

Fred Fish Disk 99

A-Render	Version .3 a Ray- Tracing Construction Set for the Amiga Computer by Brian Reed ED

Fred Fish Disk 100

Berserk	Must see animation, by Leo Schwab
Conman	Console handler replacement, provides line editing and command line histories transparent to application prog uses CON: windows. Shareware V1.0 by W Hawes. E.
WBLander	Workbench display hack game, upgrade of "Rocket" on FF85, now with sound effects. By Peter da Silva. E

Fred Fish Disk 101

CirPlane	Circular plane generator for VideoScape3D. Generates a clockwise circular polygon with the specified number of vertibles. V1.0 by T Floryan SE
IconAssembler	Change Workbench Icons with IFF-brush files by Stefan Lindahl E
Microspell	Standalone spelling checker scans text files and reports errors. 1000 common word list, 43,000 word main dictionary with multiple user dictionary support. Interfaces with MicroEMACS 3.9 with an emacs macro to step through the source file, stopping at suspect words and allowing the user to option. V1.0 by Daniel Lawrence, SED
Midi	midi library and utility set. Includes Midi monitor, routing utility, status utility, and more. by Bill Barton SED
PsIntrp	Postscript Interpreter reads and previews files on screen. by Greg Lee S(assy)E
StartUps	Three C startup file replacements for standard Astartup.obj and LStarup.obj. Options include (1) BothStartup.obj, for the WorkBench programs or CLI programs with or without command line parameters. (2) WBStartup.obj, for WorkBench programs or CLI programs that require no command line parameters. (3) CLIStartup.obj for CLI programs that require command line parameters but do not need to be WorkBench runnable. by Bryce Nesbitt SE

Fred Fish Disk 102

Dbug	Machine independent macro based C de-bugging package. Update FF41. by F Fish profiling support by Binayak Banerjee SE
Match-stuff	Heavy duty text pattern matching stuff. includes simple match text replacement capability. By Pete Goodeve
Sectorama	Recover lost or damaged data from floppy or hard disks or repair a damaged volume. by David Joiner E
SiliCon	Smart input line interpreter with window for full editing. Upgrade FF50 by P Goodeve., E
Xicon	Use icons to call up scripts containing CLI commands. V2.0 upgrade of FF31.by Pete Goodeve E

Fred Fish Disk 103

AvlTrees	Library and test prog. implement routines for creating and using trees held in memory.S.
Calc	A programmable RPN calculator.
Cref	A C cross ref. prog. S.
DosKwik	A pair of progs. allows you to save files to one or more floppies for quick loading. Doesn't store Dos format.
IntuiDos	A prog. to improve control and handling of the material on all disks in 'CLI-area' .
MFF-Update	A text import util. for MicroFiche Filer (demo on FF 89) and updates to some PD disk library datatbases.
Pack-It	Takes all files the files and dirs. on a disk & packs them into a single file. for modem.
Sol	Amiga version of solitaire.

Fred Fish Disk 104

Analyticalc	Is a large and powerful spreadsheet prog.

Fred Fish Disk 105

AsmProgs	Misc. assembly tools. Includes some S.
BasicProgs	LeastSquare solves least square probs ,graphs results,S.
Bison	A replacement for unix "yacc" command. S.
Dmouse	Another prog in the tradition of display hacks". S.
FlamKey	Allows keyboard and mouse inputs to be locked until a password is entered.
GravityWars	Game of planets,ships &black holes, v2.0 ,FF84 update.
IPo2C	A util. to write a C-lang definition to mimic the intuition pointer.S
Pere-et-Fil	Ex. of creating & using reentrant processes. S.
Record Replay	Similiar to "Journal" v2.0 update to FF95.

Fred Fish Disk 106

Funckey	Shareware function key editor, v1.1update to FF89. Source avail. from author(Anson Mah).
MoreArt	A small selection of some Amiga artwork.
QuickFlix	An IFF slideshow and cel animation prog,v0.13.
RistiNolla	A Finnish game. Also called Go-Moku. v1.0

Fred Fish Disk 107

Csh	V2.07 of Matt Dillon's csh like shell.S.
Diff	A util.,similiar to other common "diff" programs.S.
ProSuite	Provides ex. code of facilities such as FileIO Requester, XText, DoRequest. & tutorial on how to program the Amiga. Book 1.01.S
SVTools	Some useful tools. S.

Fred Fish Disk 108

AList	Dir listing prog. based on LD4 prg S
DirMaster	Disk cataloger. v1.0b, update to FF89. S.
Dots-Perfect	Printer Driver for an Epson MX80 printer with upgrade kit installed. S.
MonIDCMP	Lets you monitor the IntuiMessages that pass through an IDCMP window. Prints the message class,mouse coordinates,qualifier values. Great for debugging. S.
PrintPop	A util. to send common control settings to PRT: S.
Sectorama	Utilities to recover lost or damaged data from floppies & hard disks. v1.1, an update toFF102.
Tek	Vt100 emulator for a Tektronix 4010/4014. (V2.6) update to FF52. S.
Zoo	File archiver, like "arc". v1.24B. update to FF87

Fred Fish Disk 109

Machine	A new animation.
SimCPM	A CP/M sim,simulates 8080 along with h19 emulationS.
UUpc	Hook up your Amiga as a usenet node. S.

Fred Fish Disk 110

A68k	A 68000 assembler written in C. S.
Pdc	An optimizing C compiler for the 68000 processor. update to FF53. but not based on that code.

Fred Fish Disk 111

AmyLoad	A graphical monitor of cpu, blitter, & memory use. Includes two components; load.device,monitors system parameters, & amyload, which is the user interface & display program. by Jeff Kelley SE
AssignDev	Assigns multiple names to a given device. modified version of the original released on disk number 79. By Phillip Lindsay, mod by Olaf Seibert SE
Gauge	Continuously displays memory usage in a vertical bar graph Binary only. By:Peter da Silva
HeliosMouse	Another "sunmouse"prog. Automatically activates a window by mouse pointer V 1.1, update to FF94. By:Davide Cervone SE
Labels	Alphabetic & numeric ordered cross reference lists of defined system constants. Recommended for debugging purposes only, use the symbolic values in progs! By:Olaf Seibert
Mandel	Mandelbrot generator program, with bits & pieces of code from C. Heath & R.J. Mical. By:Olaf Seibert S
PopLife	A PopCLI type that plays life all over your screen. Lots of bits & pieces from Tomas Rokicki's blitlab & John Toebes' PopCLI. By:Olaf Seibert S

Fred Fish Disk 112

BeachBirds	Beach scene portrayed by sprites & sound 512K machine. By:Jerrold Tunnell B only.
Bully	Pushes all open screens around (thus the name "bully"). Show more than one demo at a time By:Mike Meyer S
DropShadow	Dropshadow V2.0, use with Bryce Nesbitt's Wavebench demo. B only. By:Jim Mackraz
HagenDemos	"RGB" & "Focus". RGB requires one meg, B only. By:Joel Hagen
Viacom	Latest version of viacom for use in conjuction with WaveBench demo. B only. By:Leo Schwab & Bryce Nesbitt
WaveBench	A neat screen hack, & runs on 512K machines. For more laughs, try in conjunction with Viacom or Ds (Dropshadow). Includes S. By:Bryce Nesbitt

Fred Fish Disk 113

AmiCron	Simple Unix "cron" type program,a background task uses a disk-resident table to automatically run certain tasks on a regular basis, at specific times. V 2.3, S. By:Steve Sampson, Amiga port by Rick Schaeffer
Dme	V 1.28f of Matt's text editor. A simple WYSIWYG editor for programmers. Not a WYSIWYG word processor Features : arbitrary key mapping, fast scrolling, title-line statistics multiple windows, iconify windows,etc. Update to FF93, S. By:Matt Dillon
DosDev	Example DOS device driver in Manx C. Version 1.10, includes S. By:Matt Dillon
M2Amiga	Demo of M2Amiga. A fast single pass Modula-2 compiler with editor, linker, a small set of interface & standard libraries. Compiles only small demo programs by limiting codesize & imports. Further development of the ETHZ compiler on FF24. B only. Demos with Source. By:R. Degen, C. Nieder, M. Schaub, J. Straube (AMSoft)
NoIconPos	Clears position info of any icons, allows WorkBench to pick a new place for the icon. Useful for disk & drawer icons where Snapshot rewrites the icon & the window information. Modula-2, another demo for M2Amiga By:Markus. Schaub

Fred Fish Disk 114

CDecl	English to C (and vice versa) translator for C declarations, a must for anyone except possibly the most hardcore C guru. By:Graham Ross, S
Vt100	V2.7 of vt100 terminal emulator with kermit & xmodem file transfer. Includes a few bug fixes posted to Usenet shortly after the posting of v2.7. Update to FF55. Includes S. By:Dave Wecker
WBLander	a special version of the WBLander program from FF100. Ending is unique. Effective use of sound, Includes S. By:Peter da Silva & Karl Lehenbauer

Fred Fish Disk 115

Killer	Masterful Video commercial of the Amiga, Beatles music, requires one meg of memory to run. Binary only. By:R. Wilt
Marketroid	Another devious sprite oriented demo with lots of "in" jokes. 512K required, includes S. By:Leo Schwab

Fred Fish Disk 116

Movies	A ram animation system with three different example animations; Kahnankas, Rocker, & F-15. Kahnankas & Rocker run on a 512K Amiga & show off overscan HAM mode. Includes a animation player program (movie), animation builder programs (dilbm, pilbm), & a text/graphics display program (vilbm). By:Eric Graham & Ken Offer

Fred Fish Disk 117

AMUC_Demo	A really neat horizontal scrolling demo that is a 2400 x 200 pixel 32 color IFF picture composed of digiview snapshots of members of the AMiga Users of Calgary, superimposed on a very wide picture of the Calgary Skyline. B only. By:Stephen Vermeulen & Stephen Jeans
ExP_Demo	Demo version of Express Paint 1.1., used to create the scrolling demo picture in the AMUC_Demo drawer on thi disk. B only. By:Stephen Vermeulen

Fred Fish Disk 118

Empire	Complete rewrite, in Draco, of Peter Langston's Empire . A multiplayer game of exploration, economics, war, etc, can last months.Use local keyboard or modem.V1.0, shareware, & S . By:Chris Gray, original game by Peter Langston
HAMmmm	Displays lines whose end points are bouncing around the screen, which is a double buffered HAM screen. The Y positions of the points are continuously copied into an audio waveform and played on all four channels, & the pitch of a just intoned chord is derived from the average X position of these points. JForth. Source By:Phil Burk
Stars	Based on original code by Leo Schwab, credits longer than actual demo. Runs on 512K Amiga. B only. By:Hobie Orris
WireDemo	Demonstrates the Amiga's line drawing speed. Runs on a 512K Amiga. Includes S. By:Matt Dillon

Fred Fish Disk 119

MicroEMACS	V3.9e of Daniel Lawrence's variant of Dave Conroy's microemacs. Update to FF93. Also included, for the first time, is extensive documentation in machine readable form. SE. Author: Dave Conroy, Enhanced by Daniel Lawrence

Fred Fish Disk 120

Amoeba	Clone of Space Invaders, one of the PDS games for the Amiga. B only BY: LateNight Developments
BackGammon	Graphical Backgammon (an undergraduate A.I. course project). Version 1.0, S. By: Robert Pfister
Bankn	A complete checkbook system offered by the author as shareware. Version 1.3, binary only. By: Hal Carter
EgyptianRun	"road race + hazards" type game. Version 1.1, B only, shareware, source available from author. By: Chris Hames
IconImage	Replace an old icon image with a new image, without affecting icontype, drawer data, etc. SE. By: Denis Green

Fred Fish Disk 121

BasicStrip	AmigaBASIC prog. helps to convert programs written in other forms of Basic to AmigaBASIC. By: George Trepal
DataPlot	Shareware,AmigaBASIC, plotting program. Also includes a least squares curve fit program. By: Dale Holt
Plot	Shareware 3-D AmigaBASIC graphing prog. & sample output plots. Source available via author.By: George Trepal
Stairs	AmigaBASIC prog. demos a musical illusion based upon perceptual circularity of widely spaced tones whose volumes are defined as a sinusoidal relationship to their frequency. By: Gary Cuba
Uedit	V2.3 of this nice shareware editor. With learn mode, command language, menu customization, and other user configurability and customizability features. Binary only, shareware, update toFF60. By: Rick Stiles
WBColors	Prog to change Workbench colors for progs that expect to be booted off their distribution disk but are run from a hard disk. SE. Author: Stefan Lindahl

Fred Fish Disk 122

Asteriods	Asteroid game. The images and sounds are replaceable by the end user. Anything goes! By: Rico Mariani
Iff2Pcs	Iinteractive puzzle prog. takes any IFF file with up to 16 colors, and breaks it up into squares to make a puzzle which the user can then piece together. V1.0,S. By: Ali Ozer
Names	A shareware program to create and manage mailing lists. Binary only. By: Ernie Nelson
Pr	Utility to print listings in different formats. Similar to the Unix "pr" program. Includes source By: Samuel Paolucci
PushOver	Board strategy game, AmigaBASIC. Push your pieces onto the board until yfive in a row in any direction.S. By: R.Yost
PuzzlePro	Create a puzzle from an IFF picture, which the user can then piece back together again. AmigaBASIC. V1.0, B only, shareware, source available from author. By: Syd Bolton

Fred Fish Disk 123

Arp	ARP stands for "AmigaDOS Replacement Project". Arp is an effort led by Charlie Heath of Microsmiths Inc., to replace the current DOS in a compatible fashion, so that current programs will continue to work. Arp also makes whatever improvements are possible, so that current and future programs will work better. Various authors contributed work
Car	One of Allen's entries to the Badge Killer Demo Contest. It apparently is an inside joke relating to a well known Amigan's experience with a certain highend graphics hardware manufacturer. Author: Allen Hastings

Fred Fish Disk 124

Icons	Some sample animated icons. By: L. Pfost
Tarot	AmigaBASIC Nice graphic of tarot cards. Author: L.Pfost

Fred Fish Disk 125

ElGato	Animation entry to the BK D Contest. Background music arrangement, requires Sonix to use. By Kevin Sullivans

Fred Fish Disk 126

Colour	Manipulate the colors of specific named screens, saving current color sets to data files, loading new color sets from data files, or interactively changing colors. S. By J. Russell
Dance	Two programs, "dancing polygons", are entries to the BKD Contest. They aresimilar, but demonstrate the range of colors available on the Amiga. S. By: John Olsen
HBHill	Animation entry to the BDK Contest. First known animation using the "Extra Half Brite" mode. By: Kevin Sullivan
Iconify	Subroutine creates an icon on the Amiga screen that can be subsequently dragged around, and double-clicked on. You can use this to have your programs "iconify" themselves to temporarily get out of the user's way. With source & demo program. By Leo Schwab
OnlyAmiga	Animation entry to BDK Contest. Three balls being juggled by pyramids rotating on their tops. By Iqbal Singh Hans
Suplib	Support library needed to rebuild various programs of Matt's from source, including DME, DTERM, etc. S By Matt Dillon
VCheck	V1.2 of virus detection prog. from Commodore Amiga Technical Support.. Will test for the presence of a virus in memory, or on specific disks. B only. By: Bill Koester.

Fred Fish Disk 127

Bounce	Entry for BDK Contest. Creates little dots that bounce around and mutiply. S By: Steve Hansel and Tom Hansel
Nemesis	Entry to BDK Contest. It is quite small for what it does, and won fifth place in the contest. B only. By Mark Riley
Ripples	Entries to BDK Contest. Unlike most other animations, it shows a fixed object from a moving point of view, instead of vice versa. By: Allen Hastings

Fred Fish Disk 128

Dis	68000 disassembler, written in assembler. S By Greg Lee
DropCloth	Place a pattern, a 2 bitplane IFF image or a combination of a pattern and image, into the WorkBench backdrop. Version 2.2, shareware, B By: Eric Lavitsky
LedClock	An extremely simple clock program, for interlaced screens only. S By: Ali Ozer
MRBackUp	Hard disk backup utility, does a file by file copy on AmigaDOS floppy disks. With an intuition interface & file compression. V1.3, Source. By: Mark Rinfret
Paint	Simple screen painting program, written in web. Requires web preprocessing program to rebuild from source. Includes source in web. Author: Greg Lee
PrtDriver	A printer driver for the Toshiba "3 in one" printer in its Qume (best) mode. Includes source in C and assembler. By: Rico Mariani
SDBackUp	A hard disk backup utility. CLI interface only. Does file compression. V 1.1, binary only. By Steve Drew
Sed	A clone of the Unix sed (Stream EDitor) program. Includes source. By: Eric Raymondw
Keys	A "hot-keys" program binds keyboard function keys to window manipulation functions (window activation, front to back, moving screens, etc). S By: Davide Cervone

Fred Fish Disk 129

DosKwik	A pair of programs which allow you to save files, or a group of files, to one or more floppies for quick loading, does not store files in DOS format, for speed. V2.0, update to FF103. B, Shareware. By: Gary Kemper
MRBackUp	A hard disk backup utility, does a file by file copy to standard AmigaDOS floppy disks. Includes intuition interface & file compression. V2.0 (with sources) and 2.1 (binary only, source available from author). Update of FF128. By Mark Rinfret
PaintJet	HP PaintJet printer driver from HP sources.
Patch	Two independent ports of Unix utility "patch", which applies context diffs to text file to automatically update them. Patch V 1.3 was ported to the Amiga by Rick Coupland and patchV2.0 was ported by Johan Widen. S By: Larry Wall

Fred Fish Disk 130

DirMaster	Shareware disk cataloger, V1.1, update of FF108, new features and enhancements. B only. By Greg Peters
Evo	Human evolution toy/tutorial with source.By S. Bonner
Hp	RPN calculator prog, supports calculations with binary, octal, decimal, hex, float, and complex numbers. Includes 32 registers for storing data & transcendental functions. V1.0, S By: Steve Bonner
Mach	"mouse accelerator" prog. with hotkeys, features of sun mouse, clicktofront, and popcli, a title bar clock with a bbs online charge accumulator, etc.V1.6a, S By: Brian Moats
PatEdit	A pattern editor for creating patterns to input to the Amiga SetAfPt macro call. Call sets the area fill pattern for the area filling graphics (RectFill, AreaDraw, etc). Includes source. By Don Hyde
QMan	Mandelbrot generator written partially in assem. for speed. Includes source. By Steve Bonner

Fred Fish Disk 131

Dfc	Copies disks like Maurader, but multitasks. Replaces diskcopy and format (smaller than either). Intuition interface. S By: Tomas Rokicki
HyperBase	Shareware database management system. V1.6, Binary only, source available from authors. FF58 update. By: Michael MacKenzie, Marc Mengel, & Craig Norborg
Life	A new version of Tomas's ancient Life game, with a new macro language for setting up patterns, good examples, S By: Tomas Rokicki
Mackie	A Popcli replacement that draws pretty lines on the screen in blanking mode. Includes source. Author: Software Distillery; enhancements by Tomas Rokicki
Mg1b	A version of Mg1b with an ARexx port and other improvements by Tomas Rokicki. Define macros & bind them to function keys in startup file. Includes source. Author: Various; enhancements by Rokicki
WFrags	Another version of Frags, Pops up a little window that updates occasionally. Good for developers to monitor what progs. are doing to memory. S By: Tomas Rokicki

Fred Fish Disk 132

Berserk	Animation, a "must see" for every Amiga user, and ranks with "Juggler" as a premier demo for the Amiga. The difference between this distribution, and FF100, this one includes "source". use it as an example for creating animations. Fred Fish felt it was appropriate to have at least one animation that was available at the "source code" level. Author: Leo Schwab

Fred Fish Disk 133

Conman	Shareware replacement for the standard console handler, provides line editing and command line histories completely transparent to any application program that uses CON: windows. V1.1, binary only, updateoFF100. New features include additional editing keys, fast search keys, undo key, clear history command, and more. Author: William Hawes
Crc	Two programs useful for generating 16-bit CRC listings of the contents of disks, and verifying that a given disk's files still compute to the same CRC's as listed. V1.0, binary only. By: Don Kindred
CrcLists	Complete CRC check files for FF1-128 using the Crc program included on this disk. These were made directly from Fred's master disks. Author: Fred Fish
Overscan	Patches the Intuition library so that sizable windows with MaxHeight of 200 (400 in interlace) and screens with Height of 200 (400 in interlace) will take advantage of the PAL overscan capability of Intuition V1.2. Useful only for European users who wish to run software written for the US market, without modifying the applications, but still using the additional space. S By Ari Freund

Fred Fish Disk 134

BoingThrows	50 frame HAM animation done with Sculpt-3D, and DigiPaint. The animation took about 325 hours of runtime to generate. By Marvin Landis
Browser	Workbench tool, using text-only windows, makes all files in the system accessible for executing, copying, moving, renaming, deleting, etc. Billed as a "programmers workbench". V1.2, binary only. BY Peter da Silva
Dme	V1.29 of Matt's text editor. Simple WYSIWYG editor designed for programmers. Arbitrary key mapping, fast scrolling, title-line statistics multiple windows, & ability to iconify windows. FF113 update,S. By Matt Dillon

Find	Utility searches for files that satisfy a given boolean expression of attributes, starting from a root pathname and searching recursively down through the hierarchy of the file system. Like the Unix find program. V1.0, includes source. By Rodney Lewis
Library	Demo version of a shareware program that stores textual information without regard to structure or content, and allows complicated searching for specific patterns. , B only. By: Bill Brownson
SmartIcon	Shareware Intuition objects iconifier. V1.0 is limited to iconifying windows, adds a new "iconify gadget" to each window, when clicked, iconifies the window into an icon in the ram: disk. B only, source available from author. By: Gauthier Groult

Fred Fish Disk 135

TeXF	A selection of 78 TeX fonts, with a conversion program to convert them to Amiga fonts. 22 different fonts at various sizes, ranging from 15 pixels high to more than 150 pixels. Conversion program can also be used with the fonts distributed with AmigaTeX, yielding an additional 1000+ fonts for use withother Amiga programs. V2.5, binary only. By: Ali Ozer

Fred Fish Disk 136

AsmToolBox	Assembler "toolbox" created to make interfacing between assembler programs and AmigaDOS easy. With source. By Warren Ring
Bison	A replacement for unix "yacc" command. From the GNU (GNU is Not Unix) effort. Port of the latest GNU version, by William Loftus, with the goal of preserving all of bison's current features. Includes source & test pro. "calc". By: Bob Corbett and Rich& Stallman.
Iff2Pcs	Interactive puzzle progr. takes any IFF file containing up to 16 colors, and breaks it into squares to make a puzzle the user can then piece back together again. V1.1, update of FF122, includes source. By Ali Ozer
Paste	Version of the Unix paste utility. Paste concatenates corresponding lines of the specified files into a single output line (horizontal or parallel merging) or concatenates them into alternate lines (vertical or serial merging). S. By: David Ihnat
YaBoingII	Game prog. demonstrating hardware sprite usage, including collision detection. Update of FF36. S. By: Ali Ozer, based on original by Leo Schwab
Zoo	File archiver, like "arc" in concept, but different in implementation and user interface details. Includes features that "arc" lacks (such as file/path names up to 255 characters in length). V 1.71, update of FF108. B. By: Rahul Dhesi, port by Brian Waters

Fred Fish Disk 137

Ct	Program to display images from a CT scanner, along with several interesting sample images of scans of real people, including a skull, brain, heart, and spine. Each image is 256 by 256 pixels in 2048 gray scale. The display software, a primitive user interface, is quite powerful, including functions like convolutions, averaging, laplacians, unsharp masking, edge detection, gradients, etc. Binary. by: J. Harman
JeansIcons	Miscellaneous cute icons created for AMUC's monthly newsletter disk. Submitted by Stephen Vermeulen. Author: Steve Jeans
Muncho	A cute little program which plays a digitized sound sample when you insert or remove a disk from your drive. If you don't like the sounds, you can replace them with your own. Binary only. By: Andrew Werth
Sit	Update to the Set Icon Type prog. on FF107. V1.10, includes source. Author: Stephen Vermeulen
VGad	A new gadget editor that takes two pictures of the window and its gadgets, one being the normal gadget state and the other being the fully selected state, then merges the data and converts to C source code. V1.0, binary only. Author: Stephen Vermeulen
VirusX	A boot sector virus check program that runs in the background and automatically checks all inserted disks for a nonstandard boot sector. Such disks can optionally have their boot sector rewritten to remove the virus. Includes source. Author: Steve Tibbett
VLabel	Program to print fancy customized disk labels. Combines an IFF picture and up to 50 lines of text (which may be placed arbitrarily in any font or point size) then print the result. The IFF picture can be virtually any size (up to 1008 by 1000). It will also print labels from a batch file produced by SuperBase. V1.20, binary only. By: Stephen Vermeulen

Fred Fish Disk 138

AmigaLine	A series of various technical notes for Amiga programmers. By: Byrce Nesbitt
Diff	Uses the same algorithm as the Unix diff program and also produces context diffs, suitable for use with patch. Binary only. By: Unknown (Decus C diff?)
Foreach	A simple but useful program that expands a wild card file specification and then invokes the specified command once per expanded filename, with the expanded filename as the command argument. Includes source. Author: Jonas Flygare
MacFont	A conversion tool to convert Mac fonts to Amiga fonts. Binary only. By: John O'Neill and Rico Mariani
ModulaTools	Various useful routines for those using in Modula on the Amiga. Update to FF94, S . By: Jerry Mack
Vt100	Two new versions of Dave's vt100 terminal emulator. One version, based on vt100 2.6, has been enhanced by John Barshinger to include an iconify feature, add full 132 column support using overscan, and other features (binary only). The second version is release 2.8 of the main-stream version of vt100, as enhanced and supported by Tony Sumrall. S. By: Dave Wecker

Fred Fish Disk 139

AmiCron	An enhanced and debugged version of AmiCron 2.3 from FF113. Includes source. By: Steve Sampson, Rich Schaeffer, Christian Balzer
ListScanner	A nice little utility to display all the Exec lists. Simular to Xplor utility FF73. Includes source in assembler. By: Heiko Rath
ProCalc	Simulates HP-11C programmable calculator. Both English & German versions. Shareware, B only. By: Gotz Muller
RemLib	Removes a specified library (if currently unused) or displays some info on all available libraries. Source in assembler. By: Heiko Rath
TurboBackup	A fast mass floppy disk duplicator with enforced verify mode to prevent errors. V1.0, binary only. By: Steffen Stempel and Martin Kopp
WArranger	Sends a window, identified by its name, to the front or back, without selecting it. Useful with AmiCron. Works on all screens. Includes source in assembler. By: Heiko Rath
WheelChairSim	A wheelchair simulator developed as a project for the Technical Resource Centre and the Albert Children's Hospital, to allow the matching of a wheelchair joystick to a child's handicap and allow the child to practice using the chair in a safe (simulated) environment. Binary only. Author: Unknown, submitted by Dr. Mike Smith

Fred Fish Disk 140

SBProlog	Volume 1 of the 2 volume Stony Brook Prolog (SBP) distribution, V2.3.2. This volume contains the executables and libraries. Volume 2, on FF141, contains the C and Prolog Source. By: Logic Programming Group at SUNY, Stony Brook Amiga port by David Roch & Scott Evernden

Fred Fish Disk 141

SBProlog	Volume 2 of the 2 volume Stony Brook Prolog (SBP) distribution, version 2.3.2. Volume 2 contains the C and Prolog source code. Volume 1, on FF140. By: Logic Programming Group at SUNY, Stony Brook Amiga port by David Roch and Scott Evernden
SmallC	An Amiga port of the Small-C compiler, written by Ron Cain and published in Dr. Dobb's Journal, in about 1980. Small-C is a rather small subset of the full 'C' language. It is capable of compiling itself, and other small, useful programs. Requires an assembler and linker to complete the package and produce working executables. Source and binary. By: Ron Cain. Amiga port by Willi Kusche.

Fred Fish Disk 142

Diff	Program uses same algorithm as Unix diff prog. and produces context diffs, suitable for use with patch.Same as FF138, but now includes the missing files (including source code). Author: Unknown (Decus C diff)
FracGen	Generates fractal pictures from "seeds" you create. Unlike any of the other "fractal generators", it can be used to load and display previously created fractal pics, modify existing fractals, or create your own fractals. V1.1, B. By:D.Houck
SciSubr	Scientific Subroutine Package from DECUS, ported to the Amiga to run with Absoft Fortran. A valuable resource of mathematical and statistical source code for those doing Fortran work on the Amiga. Author: Unknown; ported to the Amiga by Glenn Everhart

Fred Fish Disk 143

Rim	RIM-5 (Relational Information Manager), a full relational DBMS suitable for VERY large databases using B-Tree data storage, crude (by today's standards) user interface, but full source code is provided. RIM runs on a wide variety of systems, small and large, and produce compatible databases. Includes a built in HELP database and a programming language. Full Fortran source & documentation included. by: Various, Amiga port by Glenn Everhart

Fred Fish Disk 144

AnalytiCalc	V22-3D of Glenn Everhart's large and powerful spreadsheet program, update to FF104. Extra features "to have some pretentions of acting as an 'integrated system'. A virtual memory system supporting up to18000 columns and 18000 rows, multiple equations per cell, an outlining system, built-in cell annotation, and datafile access from any cell(s) of the sheet, plus an array of functions not present in most commercial spreadsheets. Source and documentation in arc'd form.

Fred Fish Disk 145

Csh	Modification of csh like shell to provide file name completion and argument execution. Requires ARP 1.1. Binary only, but includes diffs for the reference 2.07 source base. Author: Matt Dillon; enhancements by Johan Widen
DMouse	Versatile screen blanker, mouse blanker, auto window activator, mouse accelerator, popcli style programmable command key, pop window to front, push window to back, etc, widget. Very useful program! . V1.06, includes source. Author: Matt Dillon
Net	Link protoco provides essentially an unlimited number of reliable connections between processes on two machines, where each can be either an Amiga or a Unix (BSD4.3) machine. Works on the Amiga with any EXEC device that looks like the serial.device. Works on UNIX with tty and socket devices. Achieves better than 95% average throughput on file transfers. V1.20, includes sources for both the Amiga and Unix versions. Author: Matt Dillon
Tab	Tablature writing program, with intruments for a banjo and string guitar. Binary only. Author: Jeff deRienzo
TinyProlog	VT-PROLOG is a simple prolog interpreter provided with full source code to encourage experimentation with the PROLOG language and implementations. Version 1.1, includes source. Author: Bill and Bev Thompson

Fred Fish Disk 146

Blanker2	A screen blanking program that turns the screen black after 90 seconds of keyboard and mouse inactivity. V1.27.88, includes source. Author: Joe Hitchens
C-Light	A demo copy of a commercial ray tracing program, identical to commercial version but limited to ten objects per scene. Binary only. Author: Ronald Peterson
CrcLists	Complete CRC check files for FF129-141 and FF143-145 of the library, using the crc program from FF133. Made directly from Fred's master library. FF142 omitted due to a problem with the crc program. by: Fred Fish
DmeMacros	A set of DME macros which utilize templates to turn DME into a language-sensitive editor for C, Pascal, Modula-2, and Fortran. By Jerry Mack
MemoPad	A shareware intuition-based memo reminder program. Nicely done. V1.1, binary only. by: Michael Griebling

Fred Fish Disk 147

MicroGNUEmacs MicroGNUEmacs(MG 2b) contains many additions and enhancements since the original works by Dave Conroy (credit belongs to all contributors and Beta testers. Note: Amiga specific source code files and the document files have been archived. An executable copy of the PDS archive program "Zoo" is in the "c" directory

Fred Fish Disk 148

EFJ	"Escape from Jovi" A machine-code game featuring hi-res scrolling, large playfield, disk-based Hi-Score list, stereo sound, and multiple levels. Use a joystick in port 2 to control the ship. B, shareware ($8). By: Oliver Wagner
Fme	Nicely done map editor for the Fire-Power (tm) game. Features interlaced hi-res with intuition interface. See the "Readme.Inf" file for information on making a bootable disk. Includes source. Author: Gregory MacKay
HandyIcons	Adds a menustrip to the WorkBench window that allows you to run selected Workbench Tools by menu selection. Can be set up to provide custom environments. Current version supports only WorkBench Tools and not Projects. Binary by: Alan Rubright
Scrambler	A simple program that will encode/decode a text file into illegible gibberish, which resembles executable code, to evade prying eyes. Version 0.01, Binary only. Author: Foster Hall

Fred Fish Disk 149

AnimalSounds	A sample of digitized animal sounds along with a simple sound player. Authors: The Trumor Company, Inc. Sound Player by Don Pitts
DX-VoiceSorter	Written to be used with Jack Deckard's VoiceFiler program. (Disk 82). It allows for the sorting of a number of voicefiles stored using that program into a new voicefile of voices made up from various files. Includes source. Author: David Bouckley
Keep	A nice little utility program with an intuition interface for BBS and network junkies who download messages in one large file and then read them off-line. Using only the mouse, you can drive through such files a message at a time, examine each at your leisure and tag those you wish to keep. Version 1.2, binary only, but source available with donation to author. Author: Tim Grantham
Less	Like Unix "more", only better, with forward and backward scrolling, searching and positioning by percent of file and line number, etc. Now lets you also print the current file. Very useful! This is Amiga version 1.3, an update to the version on disk number 92. Includes source. Author: Mark Nudelman, Amiga port by Bob Leivian
Scheme	"Scheme is a statically scoped and properly tail-recursive dialect of the Lisp programming language invented by Guy Lewis Steele Jr. and Gerald Jay Sussman." Binary only. Amiga port by Ed Puckett

Fred Fish Disk 150

AirFoil	An update to the Airfoil generator on disk #71. Generates airfoil models as well as their corresponding streamline and pressure distributions. Includes source. Authors: Russell Leighton Addendum by David Foster
DC10	An AmigaBasic DC-10 instrument flight simulator. Appears to be quite in-depth with flight-planning and take-off options along with an extensive documentation file. Requires rebuilding on a separate disk and was successfully done so by following the author's instructions in the ReadMe_First file. Author: Jan Arkesteijn
ExecLib	A working example of how to build and use user-defined disk-resident libraries. Of special interest to developers working with Lattice C. Author: Alex Livshits
Iconizer	A utility program that saves your current mouse pointer to a small icon. You can restore the pointer just by double-clicking on its icon. Allows for building a whole library of pointers and to use them whenever you want. Binary only. Author: Alex Livshits
Pilot	An implementation of the PILOT language for the Amiga, including a demo done for the National Park Service. PILOT is a limited use language for use in educational and computer based instruction programs. Binary only with Beta test kit available from authors. By: T. LaGrone
StealMemBoot	A small utility designed to be a direct replacement for NoFastMem kind of programs. It modifies the boot block of a disk, so when you boot with it, all memory allocations will return only CHIP memory. Author: Alex Livshits

Fred Fish Disk 151

GlobeDemo	Graphics demo displays very smooth transitions of the rotating earth. Pop-up menu. Source. by: Bob Corwin
Icons	Yet another potpourri of interesting icons to choose from if you need one for your own program. by: Dave Turnock
Pcopy	A small intuition-based disk copier similiar to the resident "DiskCopy" except with write-verify and other user-selectable options. Useful for making multiple copies with reliable data. Requires two disk drives. S. by: Dirk Reisig
SCT	A CLI-based utility (SetColorTable) for displaying and/or setting a screen's colors. Save the colors of a screen to be restored later, or copy one screen's colors to another. Includes source. Author: aklevin
SlideShow	Very nicely done slide-show program written in assembly language. Features forward/backward presentation and creative screen wipes. Currently works only with IFF lo-res pictures. Executable only along with some new IFF pictures to have come my way. Shareware ($16). Authors: Mike McKittrick and Sheldon Templeton
Surveyor	A little utility that opens a window on the current screen and displays information about the pointer. Allows for absolute or relative measurement between two points on the screen. Very handy for precise positioning of icons and such. Includes source. Author: Dirk Reisig

Fred Fish Disk 152

Blk	A requester making tool employing various recursive algorithms including a recursive parser. It takes input text files and converts them to C-source for including as requester declarations. Includes source. Author: Stuart Ferguson
RunBack	A variant of Rob Peck's RunBackGround program from disk number 73. Allows you to start a new CLI program and run it in the background, then closes the new CLI. This version automatically searches the command-search-path to find the program. Source. By Daniel Barrett
UUCP	This is a version of uucp (Unix to Unix Copy Program) for the Amiga, along with some miscellaneous support utilities like cron, mail, and compress. Includes source. Author: Various, submitted by William Loftus

Fred Fish Disk 153

Dme	Version 1.30 of Matt's text editor. Dme is a simple WYSIWYG editor designed for programmers. It is not a WYSIWYG word processor in the traditional sense. Features include arbitrary key mapping, fast scrolling, title-line statistics multiple windows, and ability to iconify windows. Update to version on disk number 134, includes source. by: Matt Dillon
HP11	Emulates an HP11C calculator including the program mode. Features an ON/OFF button that turns the calculator into an icon that will sit and wait until you need it again. Documentation on the features is scarce, perhaps some industrious HP owner could write a small tutorial for the benefit of those that don't own an HP calculator. Binary only. Author: David Gay
HPMam	A program to manipulate settings and fonts on HP LaserJet+ printers and compatibles. Includes an Intuition interface and some sample picture files. Version 1.0, binary only, shareware. by: Steve Robb
Synthemania	An interesting, very small (and very persistent!) musical piece. If you plan on stopping it without using three fingers, you better read the document file first! Binary only. by: Holger Lubitz

Fred Fish Disk 154

Ada	An Ada Syntax checker for the amiga. Includes lex and yacc source. Author: Herman Fischer; updates by William Loftus
AssemblyDemos	A interesting group of assembly language demos for your visual and aural pleasure. Binary only. by: Foster Hall
DiskLib	Two utilities for those people who like to split up PD disks into disks of different categories. Includes source. by: Wilson Snyder
Guardian	Another virus diagnosing and vaccination program. Recognizes any non-standard bootblock. Includes a small utility program to permanently place the program on a copy of your kickstart disk in place of the seldom (if ever!) used Debug() function. Binary only. by: Leonardo Fei
PrintSpool	A print-spooling program. Very useful for printing files in the background. Many command-line options. Version 1.0.0, Includes source. Author: François Gagnon
Utilities	A group of four little utility programs: UnDelete - Undelete a file from floppy (DF0:) to any device you request, checks for a disk in the drive and allows you to abort cleanly with a CTRL 'C'. WhereIs - Looks for a file and or directory defaults to the current device CAL - Clone of the Unix CAL , dates fromyear 1 to 9999. DClock Simple title bar clock/memory gauge with pop to front.
VirusX	An update to the virus-detecting program of the same name on disk number 137. This version also checks for the Byte-Bandit strain. Version 1.21, Includes source. by: Steve Tibbett
Virus_Alert!	Yet another anti-virus program with a twist. Once installed a message is displayed just after a warm or cold boot notifying the the user that the disk and memory are virus-free, and forcing a mouse-button press before continuing. Anything writing to the bootblock thereafter will destroy the message and a normal virus- infected boot (???) will take-place. Versions 1.01 and 2.01, Binary only. by: Foster Hall
Wicon	A "Window Iconifier". Allows you to turn your windows into small icons which can be later recalled. Currently installed with MacWin to give your windows a "rubber-banding" effect. Version 1.14, Includes source. by: Steven Sweeting Introducing the Amiga

Fred Fish Disk 155

AsmExamples	A couple of assembly code examples. Executables are not extremely useful but the code might be of considerable benefit to beginning assembly languageprogrammers. by: Henrik Clausen
Bison	A replacement for unix "yacc" command. From the GNU (GNU is Not Unix) effort. Contains updates to the version on disk number 136, submitted by two separate sources. Includes source. by: Bob Corbett and Richard Stallman, updates by William Loftus & Scott Henry
NoSmoking	Sample program showing the use of a recoverable alert while displaying a personal health message. Includes source. Author: Theo Kermanidis
Scenery	A very nice assembly language random scenery generator. Generates very realistic looking landscapes. Includes intuition interface and lots of menu options. V. 1.0, binary only by; Brett Casebolt

Fred Fish Disk 156

Blocks2	Amusing and colorful display of a moving trail of"blocks". Update to version on disk number 71, however this version also Source. by: Gary Walker
Flex	Flex is a replacement for the UNIX "lex" (lexical analyzer generator) program that is faster than lex, and freelyredistributable. Includes source. Authors: Jef Poskanzer, Vern Paxson, et. al. Submissions by William Loftus and Scott Henry
Go64	Another screen hack aimed at an earlier Commodore product (Not to be confused with the commercial product Go-64! fromSoftware Insight Systems). Includes source. by: Joerg Anslik
Grammars	A group of lexical grammar files for Ada, C and Pascal for use in conjunction with the flex program on this disk and thebison program on disk #155 by: Various, submitted byWilliam Loftus
OOPS!	Tired of the monochrome background color of your Workbench or CLI? Then try this colorful screen hack to brighten things up!Includes source. by Joerg Anslik

Fred Fish Disk 157

60or80	A small utility to toggle the 60/80 column text modes without having to go through preferences. Works from either the CLI or the Workbench. S by: Mark Schretlen
AmicForm	Creates a phonebook containing only those areacodes and exchanges reachable through PC-Pursuit. Input any of Chet Solace's Finalist BBS lists and it creates the phonebook in a form usable byAmicTerm other popular terminal programs.V. 1.3, Binary only. by: J. Motsinger
AnimBalls	A nifty little animation program that allows you to create a collection of balls in three-space and then interactively rotate them in real time using the mouse. Includes source.by: Jim Guilford

BootBack	A handy little utility to copy and save the boot block from a disk, then later restore it should the disk get stomped on by some ugly virus. Source. by: David Joiner
ECPM	A CP/M emulator for the Amiga.Emulates an 8080 along with H19 terminal emulation. Update from version on disk number 109.Source.by: Jim Cathey; port by Charlie Gibbs; Significant improvements by Willi Kusche
KeyFiler	BBS message file sorter that allows sorting by keyword. Includes a textreader, Soundex matching, and limited wildcardcapabilities. V. 1.0, Binary only.by: J. Motsinger
ScreenZap	A little utility to clean away screens that are left by illbehaving programs. It will kill every screen behind the WorkBench, noting how many it gets. The screens in front of WB are not affected. Source. by: Lars Clausen
SetPrefs	Allows you to build a whole library of preference settings and instantly switch back and forth between them. Affects allpreference settings not just the colors. Very useful for machines with multiple users or multiple external devices. Includes Amiga's default and various sample preference settings. B only. by: Martin Hippele
Xicon	Xicon lets you use icons to call up scripts containing CLI commands. This is version 2.01, an update to FF102. Includes source. by: Pete Goodeve

Fred Fish Disk 158

DiskX	Nicely done Sector-based disk editor. Binary only by: Steve Tibbett
MemBoardTest	Originally designed for production testing of A1000 memory boards. Very nice intuition interface. Version 2.4, Source in Modula by: George Vokalek
MSDOS	A program to list files written in standard MS-Dos orAtari ST format. The files can then be copied to Ram and rewritten to disk in Amiga-Dos format. Binary only, Shareware, V. 0.1.Author: Frank Wibbeling
PCBTool	Early version of a shareware PC Board layout program Lots of options including variable size pads and traces, grids, grid snap, layers, zoom, selectable centering, text and more. This version does not support printer/ plotter dumps or libraries. V.2.6, B only .by: George Vokalek
ScreenX	A handy little background utility that provides a small clock/memory counter in its inactive mode and a versatile screen manipulator when called upon. Binary, source available from author, V2.1.by: Steve Tibbett
TaskX	A "real-time" task editor. Lets you list and set priorities of all currently running tasks. Binary , V. 2.0. by: S. Tibbett
VirusX	Update to FF154, checks for a couple of additional new strains. Includes source V. 1.6.by: Steve Tibbett
YachtC3	Update to FF10, contains some fixes and incorporates a simple sound process. V3, includes source. Author: Sheldon Leemon, with enhancements by MarkSchretlen

Fred Fish Disk 159

Free	A little command to put in your c directory that returns memory status and number of tasks currently served by EXEC Includes source. by: Joerg Anslik
MidiTools	A group of several different utility programs for those who run a Midi system. Binary only. by: Jack Deckard
StarChart	Nicely done intuition based program to display and identify about 600 stars, galaxies and nebulae visible in the Northern hemisphere. V.1.2, Source. by: Ray R. Larson
TaskControl	Nicely done task-handing program allowing you to put to sleep, kill or change priorities of the all the currently loaded tasks.Also potentially GURU-producing, so be careful what tasks you kill, change priorities of, etc. Handy windowizer will reduce it almostto an icon Binary only.by: J. Martin Hippele
TUC	"The Ultimate Clock". Another window title clock/memory minder. This one is in 132 columns! Also gives the free memory on drives DF0, DF1 & DF2.Includes source. by:Joerg Anslik

Fred Fish Disk 160

Calls	A little utility to help analyze the flow of a C-program by laying out the functions called in a hierarchical manner.Originally from Usenet with major revisions by Kevin Braunsdorf, Amiga portby George MacDonald
Check	A useful little utility for finding structural errors in C-source code. Many command-line options. V.1.03, binary only.by: Keith Elbertson
Dis	A 68000 disassembler, written in assembly, this is an update to the version on disk #128. Includes source by: Greg Lee with enhancements by Willi Kusche
DMouse	A versatile screen & mouse blanker, auto window activator, mouse accelerator, popcli, pop window to front, push window to back, etc, widget. V. 1.09, includes source Update to FF 145 by: Matt Dillon
DWIP	"Daisy Wheel IFF Printer". A graphics printing utility that allows the printing of IFF pictures on a daisy wheel printer. Includes source. by: Ken Van Camp
M4	A UNIX M4 look-alike macro processor intended as a front end Ratfor, Pascal, and other languages that do not have a built-in macro processing capability. Pd M4 reads standard input, the processed text is written on the standard output. by: Ozan S. Yigit (oz)
MemoPad	A shareware intuition-based memo reminder program.Nicely done. Update to version on disk #146, v .2, binary only.by: Michael Griebling
NeuralNets	A neural network example using the general-ized back-propagation delta rule for learning, specifically applied to the tabularasa Little Red Riding Hood instance. by: J. C. Hoskins

Fred Fish Disk 161

Friends	Sscreen hack with command-line options to keep your mouse pointer company when you step away. Source by: Michael Warner
Getsprite	Simple little program to convert Dpaint brushes into C-sourceBinary only. by: Michael Warner
IncRev	A handy little program that will automatically increment the revision number of a program every time it is recompiled.Binary only. by: Bryan Ford
LGZ	A Map generator/editor for the LGZ game. Not useful if you don't happen to play that game, but good source example of intuition interfacing. V.0.1 by: Lars & Henrik Clausen
Mackie	A versatile cli/macro-key initiator based on POPCLI with a unique method of "screen-blanking". I won't say more, just try it!V.1.1, with source. by: Thomas Rokicki
Nag	A shareware appointment calendar with it's own editor and a unique 'nagging' feature utilizing the Amiga's voice and audio devices. V.1.6, binary only. by: Richard Lee Stockton
Perl	Practical Extraction and Report Language, an interpreted language optimized for scanning arbitrary text files, extracting information from those text files, & printing reports based on the information.by: Larry Wall
VRTest	Another anti-virus utility that allows visual inspection of ram starting a $7E7FE, ram cleaning, bootblock inspection and vector monitoring/ reseting. Written in assembly V. 3.2, binary only.by: Babar Khan
XBoot	Very simple utility to convert a boot block into an executable file to use your favorite debugger (Wack, Dis, etc.) to study it. withsource.by: Francois Rouaix

Fred Fish Disk 162

Avi	A workalike version of the UNIX vi editor for the amiga.Thoug not especially recommended for beginners, designed for those of you who may have the vi commands permanently hard-coded into your fingertips! V.1.0, binary only. by: Peter Nestor
CLI_Utilities	This directory contains several subdirectories with small utilities, collected from various sources, only usable from the CLI. Some with source. Author: Various
Dark	A small graphics and animation demo.withsource. by: Phil Robertson
Flow2Troff	A little utility to convert from New Horizons Software "FLOW" files to UNIX "troff" files, suitable for printing on any troff-compatible laser printer. V. 1.0, includes source and a sample "FLOW" file. by: Daniel Barrett
LabyrinthII	A shareware role-playing text adventure game similar in operation to the Infocom text adventures.Includes source. by: Russell Wallace
Iffar	Maintains archives of Interchange File Format (IFF) FORM CAT and LIST files in a manner that complies with the IFF CAT specification. V.1.2, includes source by: Karl Lehenbauer
SetPALorNTSC	A couple of utility programs for testing the suitability of a developed program in either the PAL or NTSC environments. Includes source and a sample program.by: Peter Kittel
TES	"The Electronic Slave" adds a gadget strip to the top of he cli window to perform such functions as device directories info, run ED, and time. Currently, assignments are hardcoded but not difficult to change if you own a compiler. V. 1.1, with source.by: Joerg Anslik
UnknownGirl	Amall musical piece similar in execution to Synthemania" on FF153. Binary only. byHolger Lubitz

Fred Fish Disk 163.

Bankn	A complete shareware checkbook system. Update to FF120. V1.5, binary . by: Hal Carter
FiveInLine	Board-playing game similiar to Go-Moku, Ristinolla, etc. Fast-paced & addictive! Source. by: Njel Fisketjxn
MachII	A "mouse accelerator" program which includes hotkeys, the features of sun mouse, clicktofront, popcli, title bar clock with a bbs online charge accumulator, and more. Update to FF130. V2.4c, binary only. by: Brian Moats
MemTrace	Routines help debug memory allocation and freeing during program development. Complains if you try to free memory you didn't allocate & reports on memory not freed when your prog finishes. by: Jojo Wesener
PcPatch	Patches for PCCopy and PCFormat from the 'EXTRAS 1.2' disk, to allow reading/writing/formatting 3.5 inch 360k (2 sides/ 40 Tracks/9 sectors) MS-DOS disks. by: Werner Guenther
ReadmeMaster	A nifty little database for finding those programs that you know exist somewhere (???) in the AmigaLibDisk library. Maintains a key-word dictionary of the Contents descriptions that allows searching by disk number, program title, author's name, or some other descriptive word. Currently supports disks 1-154 with planned updates. Binary by: Harold Morash
View	A mouse-oriented text file reader. Sample operation is demonstrated in reading the View.doc file, instead of using the usual "Less" text file reader. by: Bryan Ford

Fred Fish Disk 164.

C-Functions	A group of four little C-functions to add to your library to make your programming life a little easier. Includes source and a small demo program showing some of the results. by: Lars Thuring
DiskSalv	Very useful program to recover files from a trashed AmigaDOS disk. Can also "undelete" files deleted by mistake, so long as they have not overwritten by further disk activity. Requires two disk drives. Many enhancements since the original version on disk #20. Version 1.3, Binary only by: Dave Haynie
Hed	A handy little editor that is more user-friendly than "Ed", yet doesn't require the memorization of complicated keystrokes of some of the larger, more powerful, editors. Binary only. by: Hal Carter
Newton	Uses the "Newtons Method" algorithm to estimate both real and imaginary roots of a polynomial of degree 20 or less. Version 1.0, includes source. by: Daniel Barrett
NewZAP	A third-generation multi-purpose file sector editing utility, from the author of FileZAP. Displays and edits full 512-byte sectors via a 106 character wide internal font. Includes a search feature to find specific strings or hex digits, forwards or backwards. Update to FF#58. Version 3.18, Binary only. by: John Hodgson
PcView	Provides the PC community with the opportunity to display IFF pictures to the best of EGA's ability. Displays Amiga pics, IBM-PC Deluxe Paint Pics, Apple II-GS Deluxe Paint Pics, and others in the IFF standard format. Includes source. Author: John Hodgson
PolyRoot	Another Polynomial root-finder using the Newtonian algorithm. Nicely done in AmigaBasic with good documentation file. Version 2.00. Author: Jon Giorgini
PrtDrivers	A couple of new Printer Drivers. Digital Equipment's LN03+ laser printer, Mannesmann Tally's MT420d dot matrix. Authors: DEC LN03 — Bernie Mentink MT420d — Sascha Wildner
Zoo	A file archiver, much like "arc" in concept, but different in implementation and user interface details. Includes some nice features that "arc" lacks (such as file/path names up to 255 characters in length). This is version 2.00, an update to FF 136. Binary only. by: Rahul Dhesi, Amiga port by Brian Waters

Fred Fish Disk 165.

Conman	Extremely useful replacement for the standard console handler, provides line editing and command line histories. Completely transparent to any application program that uses CON: windows. Shareware, and well worth a donation to the author. V1.3, binary only, update to FF133. by: William Hawes
CPM	Another CP/M emulator independently authored from the version that appeared on disk #157. Emulates a CPM computer with a Z80 processor connected to an ADM3A terminal. Assembly source included. by: Ulf Nordquist
Parsnag	A program to aid in performing color separations on Epson JX-80 printers. Source. by: John Hodgson
PlotView	A couple of programs, Plotview and Plot2Am, for viewing UNIX plot files. Also included are two sub-directories: Plot — a device independent plotting package for the Amiga, compatible with the UNIX plot subroutine package and Plot2Tek — converts UNIX plot format files to Tektronix 410x terminal graphic commands. Source included. Author: Joel Swank
RamCopy	A copy program designed for machines with 1 meg or more of Ram and only one disk drive. Copy a complete disk in only one pass. by: Stephen Gunn
SPUDclock	A simple program that uses the narrator device to speak the time at certain user specified intervals. Lots of command line options. Version 1.2, includes source. Authors: Robert E. Beaty and H. Bret Young

Fred Fish Disk 166.

AutoGraf	Collects and graphically displays information on auto mileage. Features such as miles per gallon, cost per mile, miles driven, highs, lows, averages, etc. Includes sample data file, a couple utility programs and source. Version 1.0 Author: Joel Swank
Cref	C cross referencer program. Prints out your code with line-numbers and complete key-word cross-referencing. Update fto ff#103 which had a serious bug. Includes source. by: Mike Edmonds; Amiga port by Joel Swank
MultiCalc	Yet another RPN type graphic calculator. This one generates answers with extreme precision (if 3000 digits is enough!) Features a 48-digit scrollable display, mouse driven with lots of keyboard shortcuts, & iconification during non-use. Binary . by Ken Johnson
Stevie	A public domain clone of the UNIX 'vi' editor. Supports window-sizing, arrow keys, and the help key. Version 3.10a, includes source. Amiga port by Tony Andrews

Fred Fish Disk 167.

CDecl	English to C (and vice versa) translator for C declarations. This little gem will translate english such as "declare foo as pointer to function returning pointer to array 10 of pointer to long" into "long "(*(*foo)())[10]", and vice versa. Update to FF114. Includes source. by: Graham Ross with enhancements by David Wolverton, Tony Hansen, Merlyn LeRoy, Udi Finkelstein & more.
CLIcon	Run CLI programs from workbench, similar in operation to IconExec, but more versatile. by: Bryan Ford
CloseMe	Another ingenious perversion in the screen hack category. Don't miss this one...surely destined to become a classic! Includes source. by: Charlie Gibbs
DSM	(Dynamic Sound Machine) Demo version with "Save" disabled of a program which will take any IFF sound or raw data and save it as a totally self-contained, runable program. by: Foster Hall
MRPrint	A cli-based text file printing utility with lots of nice features, including tab-to-space expansion, page headers, line numbers, margin control with line-splitting and pagination correction, ARP wildcard support, and auto- rejection of files containing binary characters. Version 3.1, Includes source. Author: Mark Rinfret
Smus3.6a	An enhanced version of the smus player that last appeared on FF58 Author: John Hodgson
Sounddemos	Some very nice demos for showing off the incredible audio power of the Amiga!. 100% assembly language, Connect the stereo for these! Author: Foster Hall

Fred Fish Disk 168.*

Fred Fish Disk 169.*

* Fred Fish 168 and Fred Fish 169 contain and programs submitted by Matt Dillion and both disks must be purchased at the same time to utilize the programs on the disk. In order to maintain the congruity of Matt Dillion's files, Fred kept the disks in their original form. We apologize for this inconvienence, but we feel the library should remain intact under Fred's original guide lines.

CONFIG V1.00	Configure friendly programs like DME for the initial window placement (configs the executable) by MDillion
Clock V1.00	Simple clock, configurable with config by MDillion.
DME V1.31	My programming oriented editor by MDillion
DMouse V1.10	Mouse enhancer .. acceleration, blanking, etc...(now handles requesters better w/ auto-activate) by M Dillion
Backup V2.01	HD backup/restore by M Dillion
SUPLIB	Support (link time) library required to compile Matt Dillion's programs. by M Dillion
LIBREF	Utility for generating run-time library link library assembly files, lvo.asm, & the run-time library's vector list. Currently very Aztec in what it generates. by MDillion
DRES V1	Support (run time) library w/ lots of functions including a generic parser and IPC. by Matt Dillion
DASM V2.11	Small-systems assembler. Handles 6502 and some of the Motorola single chip microcomputers. Generates code (not object module oriented). by Matt Dillion
FTOHEX	Part of DASM used to convert a DASM executable into an intel-hex formatted ascii file. by Matt Dillion
FILES V1.2	Matt Dillion's disk catalog program. by MDillion
SHELL V2.10	Added enviroment variable support and more. Use ConMan if you want command line editing by MDillion
FINDIT V1.00	Search for a filename (wildcard) by MDillion
LIBS V1.00	list libraries/devices in ram or attempt to remove libraries. by MDillion
SCAT V1.00	Utility to 'cat' binaries without blowing up the display by MDillion
ADDCR V1.00	Utility to add CR's before LF's in files by MDillion
REMCR V1.00	Utility to remove CR's in files by MDillion
CMP V1.00	Utility to compare two files by MDillion

Fred Fish Disk 170.

Aftterm	Commmunications program utilizing IBM 3278 terminal emulation. Binary only. by: Don Brereton
Dis6502	A ported 6502 disassembler with support added for C64 binary files. Includes source. by: Robert Bond, Amiga port by Udi Finkelstein
FastText	Blitter based fast text rendering routines written in assembly. Unique in the fact that they speed up rendering of non-proportional fonts of any height, and from 4-16 pixels in width. Source and test program included. by: Darren M. Greenwald
MRBackUp	A hard disk backup utility that does a file by file copy to standard AmigaDOS floppy disks. Includes an intuition interface and file compression. Version 2.4, Update to FF129. Binary only by: Mark Rinfret
PtrAnim	Nifty pointer animation program, includes lots of samples, a utility program and instructions on creating your own animations. Binary only, shareware by: Tim Kemp
Surf	Generates bezier surfaces of revolution. Will produce some amazing pictures of wineglasses, doorknobs, or other objects one could turn on a lathe. Includes the capacity to map IFF image files onto any surface that it can draw. Source included by: Eric Davies
Turbo	Opens a small window with a gadget that when selected, turns off bitplane, sprite, copper and audio DMA, to increase system speed. Source. by: Oliver Wagner

Fred Fish Disk 171.

AZComm	Modified version of Comm 1.34 that contains Zmodem send, receive, and resume receive. Version 1.00, Binary only. by: SS. Patel, based on Comm 1.34 by DJ James
Maze	A couple of very nice demos for the creation and use of single-solution mazes, one of which is practically a stand-alone game. Includes source. by: Werner Gunther
Sozobon-C	Atari ST version of what appears to be a full K&R freeware C-compiler, assembler and linker. The compiler main pass and the assembler were compiled and tested on an Amiga A2000 with only minimal changes, and they appear to work (to the extent that they believe they are running on an Atari-ST), so an Amiga port should be relatively easy. by: Sozobon, Limited.
Xoper	Very comprehensive program to monitor and control system activity. Monitor cpu, memory usage, ports, interrupts, devices. Close windows, screens, show loaded fonts or last Guru code number. Clean up memory, flush unused libraries, devices, fonts. etc. and a whole bunch more! Spawns its own process. A very handy background task to have loaded. Assembly source included. by: Werner Gunther

Fred Fish Disk 172.

DataToObj	A utility to convert raw data files (sprites, image data, text, etc.) directly into object code which can then be linked to the main program without the need to go through the compiling process. Includes source. by: Werner Gunther
Handshake	A full featured VT52/VT100/VT102/VT220 terminal emulator. he author has taken great pains to support the full VT102 spec. Now supports ANSI colors, screen capture and more. Update to version on FF60. Version 2.12a, binary only, shareware. by: Eric Haberfellner
MFix	Small program to insert in the startup-sequence of the commercial program, Marauder II, from Discovery Software, International. When the copy process starts, the rainbow screen is covered by a bare screen until the copy is finished. Author claims a 25% decrease in copy time is achieved. V1.0, Binary. by: Stephen Gunn
PopInfo	A "shrinkable" workbench utility to show you some info Workbench doesn't, such as free memory on external devices, chip, fast, and total ram usage and more. V2.0, Includes source. by: Jonathan Potter
ProCalc	Simulates an HP-11C programmable calculator. Lots of enhancements and bug fixes since the original version on FF139. V1.2, binary only, shareware by: Gotz Muller
Spiff	Make controlled approximations between two files. Similar to "diff" but more versatile. Allows for the handling of numerals as string literals or numeric values with adjustable tolerances. Provides for embedded commands, scriptfiles, and many other command-line parameters. Potentially very useful, needs some Amiga-specific work. Source and example files included by: Dan Nachbar, Bell Communications Research (BELLCORE)

To Be Continued......

In Conclusion

To the best of our knowledge, the materials in this library are freely distributable. This means they were either publicly posted and placed in the public domain by their authors, or they have restrictions published in their files to which we have adhered. If you become aware of any violation of the authors' wishes, please contact us by mail.

IMPORTANT NOTICE!

This list is compiled and published as a service to the Commodore Amiga community for informational purposes only. Its use is restricted to non-commercial groups only! Any duplication for commercial purposes is strictly forbidden. As a part of Amazing Computing™, this list is inherently copyrighted. Any infringement on this proprietary copyright without expressed written permission of the publishers will incur the full force of legal actions.

Any non-commercial Amiga user group wishing to duplicate this list should contact:

PiM Publications, Inc.
P.O.Box 869
Fall River, MA 02722

PiM Publications Inc. is extremely interested in helping any Amiga user groups in non-commercial support for the Amiga.

Amaze Me

Please use this order form when subscribing to Amazing Computing™, ordering Back Issues, or ordering Amiga™ Public Domain Software

Name__________________________________
Street__________________________________
City______________________St._____Zip______

PROPER ADDRESS REQUIRED. In order to expedite and guarantee your order, all large Public Domain Software orders, as well as most Back issue orders, are shipped by United Parcel Service. UPS requires that all packages be addressed to a street address for correct delivery.

Please circle the appropriate item: **New Subscription** **Renewal**

Subscription: $__________

Please start my subscription to Amazing Computing™ with the next available issue or renew my current subscription. I have enclosed $24.00 for 12 issues in the U.S. ($36.00 Canada and Mexico, $44.00 foreign surface). All funds must be in U.S. Currency on a U.S. Bank

Back Issues:
$5.00 each US, $6.00 each Canada and Mexico, $7.00 each Foreign Surface.

Please circle your Back Issue choices below:

Vol1.1 Vol1.2 Vol1.3 Vol1.4 Vol1.5 Vol1.6 Vol1.7 Vol1.8 Vol1.9 Vol2.1 Vol2.2
Vol2.3 Vol2.4 Vol2.5 Vol2.6 Vol2.7 Vol2.8 Vol2.9 Vol2.10 Vol2.11 Vol2.12 Vol3.1
Vol3.2 Vol3.3 Vol3.4 Vol3.5 Vol3.6 Vol3.7 Vol3.8 Vol3.9 Vol3.10 Vol3.11 Vol3.12

Back Issues: $__________

Public Domain Software:
$6.00 each for subscribers (yes, even the new ones!)
$7.00 each for non subscribers
(three disk minimum on all foreign orders)

Please circle your Public Domain Software choices below:

Amazing on Disk:

A#1. . .Source & Listings V3.9

Amicus:

A1 A2 A3 A4 A5 A6 A7 A8 A9 A10 A11 A12 A13 A14 A15 A16 A17
A18 A19 A20 A21 A22 A23 A24 A25 A26

Fred Fish:

FF1 FF2 FF3 FF4 FF5 FF6 FF7 FF8 FF9 FF10 FF11 FF12 FF13 FF14 FF15 FF16 FF17
FF18 FF19 FF20 FF21 FF22 FF23 FF24 FF25 FF26 FF27 FF28 FF29 FF30 FF31 FF32 FF33 FF34
FF35 FF36 FF37 FF38 FF39 FF40 FF41 FF42 FF43 FF44 FF45 FF46 FF47 FF48 FF49 FF50 FF51
FF52 FF53 FF54 FF55 FF56 FFNA FF58 FF59 FF60 FF61 FF62 FF63 FF64 FF65 FF66 FF67 FF68
FF69 FF70 FF71 FF72 FF73 FF74 FF75 FF76 FF77 FF78 FF79 FFNA FF81 FF82 FF83 FF84 FF85
FF86 FF87 FFNA FF89 FF90 FF91 FF92 FF93 FF94 FF95 FF96 FF97 FF98 FF99 FF100 FF101 FF102
FF103 FF104 FF105 FF106 FF107 FF108 FF109 FF110 FF111 FF112 FF113 FF114 FF115 FF116 FF117 FF118 FF119
FF120 FF121 FF122 FF123 FF124 FF125 FF126 FF127 FF128 FF129 FF130 FF131 FF132 FF133 FF134 FF135 FF136
FF137 FF138 FF139 FF140 FF141 FF142 FF143 FF144 FF145 FF146 FF147 FF148 FF149 FF150 FF151 FF152 FF153
FF154 FF155 FF156 FF157 FF158 FF159 FF160 FF161 FF162 FF163 FF164 FF165 FF166 FF167 FF168 FF169 FF170
FF171 FF172

(NA Denotes disks removed from the collection)

PDS Disks: $__________

Please complete this form and mail with check or money order to:

PiM Publications, Inc.
P.O. Box 869
Fall River, MA 02722-0869

Total: $__________

Please allow 4 to 6 weeks for delivery

Amazing Computing Reader Service Card

Want to know more about Amazing computing advertisers and their products? Here's yourchance! Fill out the card below (for instructions,see page 96) and drop it in the mail (U.S. only). We will process your request promptly and forward it to the advertiser.

Quick easy, and efficient—just like all services of Amazing Computing: Your Original Amiga Monthly Resource.

Amazing Computing Reader Service Card

Name____
Street____
City ____________________ St.____ Zip __________
Telephone____

A. Do you own an Amiga?
- ○ 1. Amiga 1000
- ○ 2. Amiga 500
- ○ 3. Amiga 2000
- ○ 4. Soon
- ○ 5. Not Yet
- ○ 6. Just Looking

B. What Amiga hardware product do you plan to buy next?
- ○ 1. Amiga 500
- ○ 2. Amiga 2000
- ○ 3. Memory Expansion
- ○ 4. Hard Drive
- ○ 5. IBM Emulators (sidecar or Bridgeboard)
- ○ 6. Printer
- ○ 7. Modem
- ○ 8. Music Tool
- ○ 9. Video Product
- ○ 10. Other

C. What Amiga software product do you plan to buy next?
- ○ 1. 'C' Language
- ○ 2. Forth Language
- ○ 3. Modula-2 Language
- ○ 4. Assembly Language
- ○ 5. BASIC Language
- ○ 6. Entertainment
- ○ 7. Telecommunications
- ○ 8. Spreadsheet
- ○ 9. Database
- ○ 10. Financial
- ○ 11. Video
- ○ 12. Graphics
- ○ 13. Music
- ○ 14. Other

D. Where do you buy your Amiga products?
- ○ 1. Local Amiga Dealer
- ○ 2. Discount Department Store
- ○ 3. Manufacturer
- ○ 4. Mail Order

E. Which type of articles do you like to see in Amazing Computing?
- ○ 1. 'C' Language
- ○ 2. Forth Language
- ○ 3. Modula-2 Language
- ○ 4. Assembly Language
- ○ 5. BASIC Language
- ○ 6. Game Reviews
- ○ 7. Business Reviews
- ○ 8. Hardware Product Reviews
- ○ 9. Software Product Reviews
- ○ 9. Programming How To's
- ○ 10. Business How To's
- ○ 11. Video Articles
- ○ 12. Graphics Articles
- ○ 13. Music Articles
- ○ 14. Hardware How To's
- ○ 15. PDS Updates
- ○ 16. Interviews
- ○ 17. Other

F. Which articles would you like to see more of in Amazing Computing?
- ○ 1. 'C' Language
- ○ 2. Forth Language
- ○ 3. Modula-2 Language
- ○ 4. Assembly Language
- ○ 5. BASIC Language
- ○ 6. Game Reviews
- ○ 7. Business Reviews
- ○ 8. Hardware Product Reviews
- ○ 9. Software Product Reviews
- ○ 9. Programming How To's
- ○ 10. Business How To's
- ○ 11. Video Articles
- ○ 12. Graphics Articles
- ○ 13. Music Articles
- ○ 14. Hardware How To's
- ○ 15. PDS Updates
- ○ 16. Interviews
- ○ 17. Other

G. Are you a subscriber to Amazing Computing?
- ○ 1. Yes
- ○ 2. No

101

January card Valid until 3/31/89
see page 112 for instructions

101	102	103	104	105	201	202	203	204	205
106	107	108	109	110	206	207	208	209	210
111	112	113	114	115	211	212	213	214	215
116	117	118	119	120	216	217	218	219	220
121	122	123	124	125	221	222	223	224	225
126	127	128	129	130	226	227	228	229	230
131	132	133	134	135	231	232	233	234	235
136	137	138	139	140	236	237	238	239	240
141	142	143	144	145	241	242	243	244	245
146	147	148	149	150	246	247	248	249	250
151	152	153	154	155	251	252	253	254	255
156	157	158	159	160	256	257	258	259	260
161	162	163	164	165	261	262	263	264	265
166	167	168	169	170	266	267	268	269	270
171	172	173	174	175	271	272	273	274	275
176	177	178	179	180	276	277	278	279	280
181	182	183	184	185	281	282	283	284	285
186	187	188	189	190	286	287	288	289	290
191	192	193	194	195	291	292	293	294	295
196	197	198	199	200	296	297	298	299	300

Amazing Computing Reader Service Card

Name____
Street____
City ____________________ St.____ Zip __________
Telephone____

A. Do you own an Amiga?
- ○ 1. Amiga 1000
- ○ 2. Amiga 500
- ○ 3. Amiga 2000
- ○ 4. Soon
- ○ 5. Not Yet
- ○ 6. Just Looking

B. What Amiga hardware product do you plan to buy next?
- ○ 1. Amiga 500
- ○ 2. Amiga 2000
- ○ 3. Memory Expansion
- ○ 4. Hard Drive
- ○ 5. IBM Emulators (sidecar or Bridgeboard)
- ○ 6. Printer
- ○ 7. Modem
- ○ 8. Music Tool
- ○ 9. Video Product
- ○ 10. Other

C. What Amiga software product do you plan to buy next?
- ○ 1. 'C' Language
- ○ 2. Forth Language
- ○ 3. Modula-2 Language
- ○ 4. Assembly Language
- ○ 5. BASIC Language
- ○ 6. Entertainment
- ○ 7. Telecommunications
- ○ 8. Spreadsheet
- ○ 9. Database
- ○ 10. Financial
- ○ 11. Video
- ○ 12. Graphics
- ○ 13. Music
- ○ 14. Other

D. Where do you buy your Amiga products?
- ○ 1. Local Amiga Dealer
- ○ 2. Discount Department Store
- ○ 3. Manufacturer
- ○ 4. Mail Order

E. Which type of articles do you like to see in Amazing Computing?
- ○ 1. 'C' Language
- ○ 2. Forth Language
- ○ 3. Modula-2 Language
- ○ 4. Assembly Language
- ○ 5. BASIC Language
- ○ 6. Game Reviews
- ○ 7. Business Reviews
- ○ 8. Hardware Product Reviews
- ○ 9. Software Product Reviews
- ○ 9. Programming How To's
- ○ 10. Business How To's
- ○ 11. Video Articles
- ○ 12. Graphics Articles
- ○ 13. Music Articles
- ○ 14. Hardware How To's
- ○ 15. PDS Updates
- ○ 16. Interviews
- ○ 17. Other

F. Which articles would you like to see more of in Amazing Computing?
- ○ 1. 'C' Language
- ○ 2. Forth Language
- ○ 3. Modula-2 Language
- ○ 4. Assembly Language
- ○ 5. BASIC Language
- ○ 6. Game Reviews
- ○ 7. Business Reviews
- ○ 8. Hardware Product Reviews
- ○ 9. Software Product Reviews
- ○ 9. Programming How To's
- ○ 10. Business How To's
- ○ 11. Video Articles
- ○ 12. Graphics Articles
- ○ 13. Music Articles
- ○ 14. Hardware How To's
- ○ 15. PDS Updates
- ○ 16. Interviews
- ○ 17. Other

G. Are you a subscriber to Amazing Computing?
- ○ 1. Yes
- ○ 2. No

102

January card Valid until 3/31/89
see page 112 for instructions

101	102	103	104	105	201	202	203	204	205
106	107	108	109	110	206	207	208	209	210
111	112	113	114	115	211	212	213	214	215
116	117	118	119	120	216	217	218	219	220
121	122	123	124	125	221	222	223	224	225
126	127	128	129	130	226	227	228	229	230
131	132	133	134	135	231	232	233	234	235
136	137	138	139	140	236	237	238	239	240
141	142	143	144	145	241	242	243	244	245
146	147	148	149	150	246	247	248	249	250
151	152	153	154	155	251	252	253	254	255
156	157	158	159	160	256	257	258	259	260
161	162	163	164	165	261	262	263	264	265
166	167	168	169	170	266	267	268	269	270
171	172	173	174	175	271	272	273	274	275
176	177	178	179	180	276	277	278	279	280
181	182	183	184	185	281	282	283	284	285
186	187	188	189	190	286	287	288	289	290
191	192	193	194	195	291	292	293	294	295
196	197	198	199	200	296	297	298	299	300

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 36 FALL RIVER, MA

POSTAGE WILL BE PAID BY ADDRESSEE

Amazing COMPUTING
P.O. Box 869
Fall River, MA 02722-0869

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 36 FALL RIVER, MA

POSTAGE WILL BE PAID BY ADDRESSEE

Amazing COMPUTING
P.O. Box 869
Fall River, MA 02722-0869

Do It NOW!!!

Subscribe TODAY and save over 49% on Amazing Computing!

Amazing Computing is available to United States subscribers at $24.00 for 12 issues. That's a savings of $23.40 off the newsstand price. Canada and Mexico, $36.00. Foreign Surface to all other countries, $44.00 (Air mail and first class rates available upon request.)

Enter my 12 issue subscription today!

Name ______________________________
Street ______________________________
City ________________ St.____ Zip ________

Amount Enclosed __________

Sorry, we cannot accept credit cards or requests for billing at this time.
All funds must be in U.S. Currency drawn on a U.S. Bank.
Please allow four to six weeks for processing.

Clip and $AVE !!

(over 49% of the newsstand price!)

Save Time and Money!

Subscribe TODAY and save over 49% on Amazing Computing!

Amazing Computing is available to United States subscribers at $24.00 for 12 issues. That's a savings of $23.40 off the newsstand price. Canada and Mexico, $36.00. Foreign Surface to all other countries, $44.00. (Air mail and first class rates available upon request.)

Enter my 12 issue subscription today!

Name ______________________________
Street ______________________________
City ________________ St.____ Zip ________

Amount Enclosed __________

Sorry, we cannot accept credit cards or requests for billing at this time.
All funds must be in U.S. Currency drawn on a U.S. Bank.
Please allow four to six weeks for processing.

Please Mail to:

Amazing
COMPUTING
P.O. Box 869
Fall River, MA 02722-0869

Place this order form in an envelope with your Check or Money Order.

Clip and $AVE !!

(over 49% of the newsstand price!)

Please Mail to:

Amazing
COMPUTING
P.O. Box 869
Fall River, MA 02722-0869

Place this order form in an envelope with your Check or Money Order.

TAX BREAK™

Tax Preparation and Planning Tool

TAX BREAK™ is easy-to-use with on screen representation of forms and schedules, fast recalculation, and constant status of taxes due or refunds. Full integration of all forms and schedules allows constant updating of tax status for evaluation of different tax scenarios.

- Scrollable, on screen look-a-like forms and schedules
- Schedules can be opened from a menu or by double clicking on the schedule result line in form 1040 in US version or T1 for Canadian version
- Ongoing status of taxes due and refunds at bottom of each form, or expanded status screen available with mouse click or F2 key
- Instant updating of forms and schedules with each recalculation
- Line by Line data entry prompts
- On screen IRS booklet
- Prints IRS approved facsimile forms
- Pop up calculator
- Zoom in and out of interlace at will

US version 1.0 includes Form 1040, Schedules A-E, SE, 1906, 2441, IRA and more!

U.S. version 1.0
$79.95 Suggested Retail
contact:
Oxxi inc.
P.O. Box 90309
Long Beach, CA 90809
(213) 427-1227
California Residents add 6.5%

Canadian version 3.0
Known in Canada as Amiga-Tax™
$69.95 Suggested Retail
contact:
DATAMAX RESEARCH
Box 5000, RR4
Bradford, Ontario L3Z 2A6
(416) 775-2225
Ontario Residents add 8%

Canadian version 3.0 includes T1, All Schedules, all Provinces and "Quebec Forms" and more! Updates available for $29.95 direct from Datamax Research

Dealers call: Southern Technology, American Software, Micro-Pace, Triangle or Air-Stat of Canada

See your local Amiga Dealer.

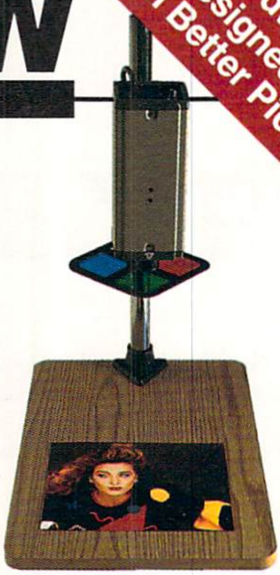
Amiga is a trademark of Commodore-Amiga, Inc.
TAX BREAK is a trademark of Oxxi, Inc.
Circle 75 on Reader Service card.

DIGI-VIEW GOLD

All-New! Hardware and Software Designed for A500/2000—Even Better Pictures Than Before.

1.

2.



Actual 4096 color Digi-View Gold picture

3. Simply the Best.

The all new Digi-View Gold is the best video digitizer for the Amiga. Period. Nothing else even comes close. Why? The secret is that Digi-View Gold captures 2.1 million colors in memory, giving you an incredible 100,000 apparent colors on screen simultaneously.

And it's easy to use. Just focus your video camera on any object or picture, and in seconds Digi-View Gold turns it into Amiga graphics that glow with vibrant color and clarity. Whether you are creating graphics for desktop publishing, presentations, video, or just for fun, Digi-View Gold gives you dazzling images with amazing simplicity.

Digi-View Gold is designed specifically for the Amiga 500 and 2000, and plugs directly into the parallel port. Digi-View Gold's powerful image capture and manipulation software (version 3.0) now has complete control of color and sharpness, full overscan, extra halfbrite, and a special line art mode for desktop publishing.

Only Digi-View Gold:

- Can digitize in all Amiga resolution modes from 320x200 up to 768x480 (full hi-res overscan)
- Uses 2 to 4096 colors (including extra halfbrite)
- Uses exclusive Enhanced HAM for super fine detail
- Is 100% IFF compatible and works with any graphics software
- Can digitize 21 bits per pixel (2.1 million colors) for the highest quality images possible
- Has advanced dithering routines that give an apparent 100,000 colors on screen simultaneously
- Has powerful Image processing controls for complete IFF picture manipulation

If you want the highest quality graphics for your Amiga, as easy as 1, 2, 3; then you need the new version of the best selling video digitizer of all time: Digi-View Gold.

Only $199.95

Digi-View Gold is available now at your local Amiga dealer.
Or call 1-800-843-8934

NEWTEK
INCORPORATED

*Requires standard gender changer for use with Amiga 1000. Video camera required; not included. NewTek sells a video camera, copy stand, and the Digi-Droid automated filter wheel for Digi-View Gold. If your local retailer doesn't carry these products, call us at 913-354-1146. Digi-View Gold is a trademark of NewTek, Inc. Amiga is a trademark of Commodore-Amiga, Inc. Be seeing you!!